

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication :
(A n'utiliser que pour les
commandes de reproduction).

2 269 150

A1

**DEMANDE
DE BREVET D'INVENTION**

(21)

N° 74 14489

(54) Procédé et dispositif de communication d'information entre des processus d'un système informatique.

(51) Classification internationale (Int. Cl.²). G 06 F 13/00.

(22) Date de dépôt 25 avril 1974, à 15 h 59 mn.

(33) (32) (31) Priorité revendiquée :

(41) Date de la mise à la disposition du public de la demande B.O.P.I. — «Listes» n. 47 du 21-11-1975.

(71) Déposant : COMPAGNIE HONEYWELL BULL, résidant en France.

(72) Invention de : John J. Bradley, Benjamin Franklin, David Slosberg, Marc Appell, Jean-Claude Cassonnet, Philippe de Rivet de Sabatier.

(73) Titulaire : *Idem* (71)

(74) Mandataire : G. Korsakoff, Compagnie Honeywell Bull, 94, avenue Gambetta, 75020 Paris.

La présente invention se rapporte d'une manière générale aux systèmes électroniques de traitement de données et a plus spécialement pour objets un procédé et un dispositif de communication d'information, entre des entités appelées processus. La notion de processus telle qu'elle est utilisée dans ce brevet est une notion complexe car bien que se rapportant à l'exécution séquentielle, linéaire d'une série d'instructions elle recouvre aussi bien le micrologiciel que les moyens matériels nécessaires à l'exécution de la séquence d'instructions considérée. Un processus est l'exécution d'un programme c'est-à-dire que ce concept inclut les réactions du micrologiciel et du matériel aux sollicitations des instructions du processus.

Un autre objet de la présente invention est de permettre l'introduction dans un système informatique d'un déroutement asynchrone des processus c'est-à-dire d'un déroutement tel que un processus en cours d'exécution A puisse alerter ou avertir un processus B sans que le processus B soit mis en attente d'un autre événement. L'opération dite de déroutement quoique très voisine de l'interruption ne doit pas être confondue avec elle. En effet une interruption permet d'interrompre provisoirement le déroulement d'un programme en cours pour faire exécuter un programme considéré comme prioritaire. Plus précisément, le traitement d'une interruption, consiste généralement dès la fin de l'instruction en cours à sauvegarder l'état de la machine puis à forcer dans le compteur d'instruction l'adresse de la première instruction associée à l'interruption. Ce programme se termine par une instruction qui restaure l'état de la machine au moment où l'interruption a été prise en compte ce qui a pour effet de permettre la reprise du programme interrompu.

Au contraire, dans un déroutement le programme ne revient pas au point où il a été interrompu mais dans un endroit quelconque du programme qui est défini par le déroutement.

Un autre objet de la présente invention est de définir un moyen de communication qui n'interrompt le processus que dans des niveaux de privilèges inférieurs à un niveau donné. Dans un système auquel peut s'appliquer l'invention sont définis : d'une part des niveaux de privilèges matérialisés par des anneaux (voir la demande de brevet française N° 73 42706 déposée le 30.11.73 pour "Protection de l'information dans un système de multitraitement de l'information mettant en oeuvre un concept d'anneaux pour représenter les différents niveaux de privilèges entre processus" au nom de la demanderesse), et d'autre part des niveaux de priorité permettant dans le temps une gestion optimisée des opérations à effectuer. Comme il sera vu par la suite, le numéro d'anneau définit quantitativement la nature de l'opération à effectuer. Grosso modo, les opérations effectuées en anneau zéro

sont des fonctions du système proprement dit qui ne peuvent en aucun cas être interrompues puisque l'on risquerait de ne plus savoir quel est l'état de la machine après l'interruption.

Le dispositif objet de la présente invention prend en compte le niveau
5 de privilège dans lequel sont effectuées les opérations.

Un dispositif de communication d'information s entre les processus a déjà été décrit dans le demande de brevet française N° 7342711 du 30.11.73 intitulée "Dispositif de synchronisation de processus par sémaphores pour un système informatique" déposée au nom de la demanderesse. Dans ce brevet,
10 la communication entre les processus est assurée au moyen de "sémaphores" qui sont des structures de données contenues en mémoire centrale ou virtuelle et qui permettent d'associer un processus et des informations nécessaires à ce processus. Deux types d'opérations ont été définis dans ce brevet, l'opération P au cours de laquelle un processus est Preneur d'informations et
15 une opération V au cours de laquelle un processus est Vendeur ou donneur d'informations.

permet de réaliser une opération

Le dispositif objet de la présente invention /du type P par des moyens différents. Les informations relatives à un processus sont contenues en mémoire centrale dans ce qu'on appelle un bloc de contrôle de processus PCB.
20 Un tel bloc de contrôle est décrit dans la demande de brevet français N° 73 42689 du 30.11.73 ayant pour titre "Bloc de contrôle de processus" déposée au nom de la demanderesse, les informations du PCB sont contenues dans des mot de trente deux bits soit quatre octets.

Selon la présente invention le procédé de communication d'informations
25 entre deux processus dans un système informatique multiprogrammé comprenant une pluralité de processus dont l'un est "en cours" et dont les autres sont dans les états "attente", "prêt" ou "suspendu" et comprenant au moins une mémoire centrale, un processeur centrale (CPU) communiquant avec ladite mémoire, ledit processeur central étant sous la commande du processus
30 "en cours d'exécution", un système d'exploitation permettant de créer ou de supprimer des processus, chaque processus étant représenté dans la mémoire centrale par un bloc de contrôle de processus, est caractérisé en ce que le mot principal du bloc de contrôle de processus contient un octet de priorité, une position de bit dudit octet indiquant qu'un déroutement doit être effectué,
35 un champ dudit octet indiquant dans quels numéros d'anneaux ce déroutement est possible, des moyens permettant la lecture du bit et du champ au début de chaque passage d'un processus de l'état prêt à l'état en cours provoquant le déroutement.

Les éléments nouveaux qui sont caractéristiques de l'invention sont énoncés
40 plus particulièrement dans les revendications annexées. L'invention elle-même tant en ce qui concerne son organisation et son fonctionnement.

que d'autres buts et avantages qu'elle présente sera plus facilement comprise en se référant à la description ci-après en combinaison avec les dessins annexés dans lesquels :

La Fig. 1 est un schéma bloc /^{de} système de multiprogrammation utilisant l'invention ;

La Fig. 2 est une représentation schématique de diverses structures du matériel utilisées par l'invention ;

La Fig. 3 est une légende des termes utilisés pour les zones de stockage réservées dans les registres représentés sur la Fig. 2 ;

La Fig. 4 est une représentation schématique d'un bloc de commande de processus ;

La Fig. 5 est une représentation schématique d'un système utilisé pour adresser un bloc de commande de processus ;

La Fig. 6 est une représentation schématique de la base du système de l'invention ;

Les Fig. 7A et 7B représentent schématiquement respectivement un segment de pile et un élément de pile ;

La Fig. 8 est une représentation schématique d'un système utilisé pour adresser les segments G et en particulier la file d'attente des processus contenue dans le segment GO ;

La Fig. 9 est une représentation schématique éclatée d'un segment GO montrant la file d'attente des processus et les liaisons de processus ;

Les Fig. 10a à 10l sont des schémas-blocs des structures du bloc de commande de processus (PCB) ;

Les Fig. 11a à 11r sont des schémas - blocs des structures de la base du système ;

La Fig. 12 est une représentation schématique des moyens utilisés pour l'adressage des segments des utilisateurs et du système en utilisant les structures de la base du système et des blocs de commande de processus (PCB).

Les Fig 13a₁- a₂ ; 13b₁- b₂ et 13c constituent une représentation schématique de l'unité de commande ;

Les Fig. 14a₁ à 14a₅ sont des organigrammes de l'unité distributeur en microprogrammée.

L'invention est utilisée typiquement dans le cadre du système matériel décrit ci-après coordonné par un système d'exploitation matériel / microprogrammée / logiciel. Sur la Fig. 1 à laquelle on se référera, les sous-systèmes de l'ordinateur comprennent le sous-système processeur 101 le sous-système de stockage 102 ou mémoire centrale et un ou plusieurs - jusqu'à 32 - sous-systèmes périphériques 103. Le sous-système de trai-

tement comprend une unité centrale (CPU) 104 jusqu'à quatre unités de contrôle Entrée-Sortie (IOC) 105. Chaque sous-système périphérique est constitué par une unité de contrôle de périphériques (PCU) 106, un certain nombre d'adaptateurs de dispositifs (DA) 107 et jusqu'à 256 périphériques E/S 108.

- 5 Le sous-système de stockage 102 comprend de un à quatre modules de mémoire à semi-conducteur contenant chacun de 32000 à 512 000 octets.

I. SOUS-SYSTEME PROCESSEUR.

- Dans le sous-système processeur 101, l'unité centrale 104 exécute les opérations de traitement fondamentales pour le système et établit la
10 jonction avec la mémoire 102. Le contrôleur E/S 105 commande tous les échanges d'informations entre le sous-système du stockage 102 et les dispositifs périphériques 103.

A. UNITE CENTRALE (CPU).

- L'unité centrale comporte un synchroniseur de mémoire principale 109
15 une mémoire tampon 110, divers éléments qui comprennent l'unité de calcul 111, des émulateurs 112 de mémoire tampon 110, un synchronisateur de mémoire principale 109, une mémoire morte 130, une unité de contrôle d'adresse 131 et une unité de mémoire associative 131. Le synchronisateur de mémoire principale 109 résout les conflits pour l'utilisation de la mémoire principale entre
20 l'unité de calcul 111, la mémoire tampon 110 et le contrôleur E/S 105. Les conflits sont résolus sur la base de la priorité ci-après ; le contrôleur E/S a la plus forte priorité suivi des opérations d'écriture en mémoire (de l'unité de calcul) et des opérations de lecture en mémoire (avec transfert dans la mémoire tampon). L'unité centrale comporte également l'unité de contrôle
25 d'adresse 131, qui commande l'adressage de la mémoire principale, et la mémoire associative 132, utilisée pour stocker les adresses de la mémoire principale les plus récemment utilisées. La mémoire tampon 110 est une mémoire tampon rapide de petite capacité qui reproduit une région sélectionnée de la mémoire principale et fait la jonction avec l'unité de calcul pour réduire le
30 temps d'accès moyen à la mémoire. Au cours de chaque cycle de lecture en mémoire, la mémoire tampon et la mémoire principale sont toutes deux accédées. Si l'information à extraire est déjà dans la mémoire tampon, la lecture en mémoire principale est interrompue et les informations sont extraites de la mémoire tampon. Autrement dit, la mémoire principale 102 est lue.
35 Chaque fois qu'une telle lecture est effectuée, l'unité centrale 101 extrait 32 octets qui contiennent l'information désirée. Cette information reste dans la mémoire tampon en vue des références futures à la mémoire. Etant donné que la mémoire tampon est transparente au logiciel, le programme contrôlant l'ordinateur à un moment donné quelconque ne peut pas déterminer si l'information qu'il traite a été extraite de la mémoire tampon ou de la mémoire /
40 principale.

L'unité de calcul 111 exécute toutes les opérations de traitement de données et de génération d'adresse à l'intérieur de l'unité centrale. Une mémoire morte de commande 130 prévue dans l'unité de calcul (cf : le livre Microprogramming : Principles and Practices, Samir S. Husson Prentice Hall Inc) contient la microprogrammerie qui initialise le système, commande l'unité centrale 104 et le contrôleur E/S 105 et décode un jeu d'instructions (non représenté). Facultativement, la mémoire morte peut fournir des instructions scientifiques, des routines d'essai, des ensembles de programmes d'émulation ou des caractéristiques spécialisées qui étendent les capacités du sous-système processeur.

Facultativement, l'unité centrale permet l'émulation des systèmes autres que le présent système. Les émulateurs 112 sont constitués par des éléments de microprogrammerie, de logiciel et, dans certains cas, de matériel.

B. CONTROLEUR ENTREE-SORTIE.

La partie du sous-système processeur constituée par le contrôleur E/S105 établit une voie de données entre un sous-système périphérique 103 quelconque et le sous système de stockage 102. Cette voie de données permet la mise en route des commandes périphériques et commande les transferts de données résultants. Un contrôleur E/S peut contrôler jusqu'à 32 unités de contrôle de canal (non représentés).

C. SOUS-SYSTEMES PERIPHERIQUES.

Dans un sous-système périphérique 103(Fig. 1) l'unité de contrôle de périphérique (PCU)106 est un processeur microprogrammé indépendant qui soulage la charge de l'unité centrale 104 en commandant les dispositifs E/S108 au cours des opérations d'entrée-sortie. L'unité de contrôle de périphérique 106 remplit cette fonction en exécutant des instructions contenues dans un programme de canal. Ce programme a pour résultat l'exécution d'opération arithmétique, logique, de transfert, de décalage et de branchement dans l'unité de contrôle de périphériques. Il existe plusieurs types d'unités de contrôle de périphériques, en fonction du type de dispositif que chacune commande à savoir, les unités d'enregistrement classiques, les mémoires de masse (à disques), les unités à bande magnétique, les systèmes de communication, etc..

Les adaptateurs de dispositifs 107 servent d'intermédiaire entre chacune des unités de contrôle de périphériques et les dispositifs qu'elle commande. Chaque adaptateur contient la microprogrammerie et la logique nécessaires pour réaliser les communications avec un type particulier de dispositif. Selon le type, un adaptateur 107 commande un ou plusieurs dispositifs.

Les principales fonctions remplies par un sous-système périphérique 103 sont les suivantes :

1. transformer les instructions de l'unité centrale en une série

de commandes acceptables par le dispositif périphérique approprié.

2. condenser et décondenser les données sous la forme requise par l'unité centrale ou par le dispositif périphérique approprié.

3. maintenir l'unité centrale informée de l'état du sous-système et des dispositifs sous son contrôle.

4. lancer et traiter indépendamment les procédures d'erreur et de rétablissement à la suite d'erreurs.

5. permettre le diagnostic direct d'un dispositif sans perturber les capacités de partage entre les dispositifs du processeur périphérique associé.

Une unité de contrôle de périphérique résout les conflits pour l'accès à la mémoire principale entre les dispositifs qui lui sont rattachés. Pendant que le contrôleur E/S résout les conflits entre les unités de contrôle de périphériques.

15 D. SOUS-SYSTEMES DE STOCKAGE.

Chaque module de mémoire 1 à 4 a une largeur de 4 à 8 octets. Le nombre de modules, leurs dimensions, et la largeur de la voie de données peuvent varier selon les dimensions de l'ordinateur. Les modules de mémoire sont imbriqués à quatre voies d'une manière telle que les quatre modules sont accédés séquentiellement (le module 1 contient les 8 premiers octets, le module 2, les 8 octets suivants, etc.....). L'imbrication ou entrelaçage réduit le nombre de conflits pour l'accès à la mémoire et réduit ainsi le temps d'accès moyen à la mémoire. La mémoire peut être remaniée en cas de défaillance, c'est-à-dire que des blocs de mémoire contenus dans un module peuvent être retirés sans destruction de l'adressage continu.

La mémoire principale 102 est constituée par un milieu de stockage capacitif sous forme de microplaquettes MOS (métal - oxyde-semi-conducteur). Ce milieu fonctionne sur le principe de la régénération pour conserver les informations. Chaque emplacement de mémoire est régénéré au moins une fois toutes les millisecondes. La conception du système assure que peu de conflits se produisent entre les temps de régénération et les accès à la mémoire (en cas de conflit, la régénération a priorité).

Une zone située au début de la mémoire principale est réservée au matériel et à la microprogrammerie. La limite supérieure de cette zone est définie par le contenu d'un registre d'adresse limite BAR (qui sera décrit ci-après) qui est visible par le logiciel. Le contenu du registre BAR est fixé au moment de l'initialisation du système. La zone de mémoire au-dessous de l'adresse spécifiée dans le registre BAR peut contenir des tables de commande Entrée-Sortie (IOC) qui définissent la configuration des sous-systèmes périphériques, la microprogrammerie de commande de l'unité

centrale (CPU) ou des microprogrammes et des tables pour l'émulation. La taille de la zone située au-dessous de l'adresse spécifiée dans le registre BAR dépend de la configuration du système. Le fait que les microprogrammes soient dans la mémoire principale ou dans la mémoire morte (mémoire de commande) dépend de la configuration du système et des applications exécutées par le système.

II. STRUCTURES DE BASE DE LA MACHINE

Il existe typiquement trois structures de données de base utilisées dans ce matériel, les formats de données, les registres visibles au logiciel et les formats d'instructions.

A. FORMATS DE DONNEES.

Les informations sont transférées entre la mémoire et l'unité centrale par multipléts de 8 bits en parallèle. Chaque unité d'information de 8 bits est appelée un octet. Des données de parité ou de correction d'erreur sont également transférées avec les données mais elles ne peuvent être modifiées par le logiciel. En conséquence, dans cette description, le terme données exclus les données de parité ou de correction d'erreurs associées.

B. OCTETS.

Les bits à l'intérieur d'un octet sont numérotés de 0 à 7 de gauche à droite. Les octets sont traités séparément ou en groupes. Deux octets constituent un demi-mot, 4 octets un mot, 8 octets un mot double, 16 octets un mot quadruple. Ce sont les formats de base utilisés pour toutes les données, y compris les instructions.

C. REPRESENTATION DES DONNEES.

Toutes les données sont sous forme binaire mais peuvent être interprétées comme données binaires, décimales ou alphanumériques. Les bits de données sont interprétés par groupes de quatre en tant que données en décimal codé-binaire, par groupes de huit en tant que données alphanumériques ou par groupes de 16 à 64 en tant que chiffres binaires. Ces derniers sont interprétés en tant que nombres signés, à virgule fixe ou flottante, en notation binaire. Tout nombre de bits contigus jusqu'à la valeur d'un double mot peut être manipulé comme une chaîne. Le jeu des caractères alphanumériques est représenté en code EBCDIC. Le code ASCII est utilisable comme autre code d'échange.

D. ADRESSE D'OCTETS.

Les emplacements d'octets dans la mémoire principale sont numérotés à la suite en partant de zéro. Chaque numéro est l'adresse d'un octet. Un groupe d'octets consécutifs est dit aligné sur un demi-mot, un mot, un mot double ou un mot quadruple, si l'adresse de l'octet de gauche du groupe est respectivement un multiple de 2, de 4, de 8 ou de 16. Chaque fois qu'un

demi-mot , un mot , un mot double ou un mot quadruple est ainsi aligné, cette unité peut être extraite à cette adresse. L'emplacement des données dans la mémoire principale est spécifié par un descripteur de données qui est accédé indirectement au cours du développement d'adresse (cf demande de brevet

5 N°73 17765 du 16 Mai 1973 intitulée : "Dispositif de développement et de calcul d'adresse dans une mémoire segmentée" au nom de la demanderesse.

E. REGISTRES VISIBLES.

Il y a 33 registres visibles aux utilisateurs dans l'unité centrale 104 (Fig. 1) dont les contenus définissent collectivement l'état de l'unité centra-

10 le . Ces registres sont de quatre types(cf. Fig. 2) :

1. Registres généraux
2. Registres de base
3. Registres scientifiques (facultatifs)
4. Registres divers.

15 F. REGISTRES GENERAUX.

Les registres généraux (GR) 201 sont utilisés pour manipuler les nombres binaires en virgule fixe et les chaînes de bits. Il existe typiquement 16 registres généraux de 32 bits dans l'unité centrale 104, désignés respectivement GRO à GR15. Les registres généraux GR8 à GR15 sont également utilisables
20 comme registres d'index. Lorsqu'ils sont utilisés comme registres d'index, ils sont appelés ici registres XO à X7. L'indexation est effectuée en utilisant le nombre entier en complément à deux de 32 bits contenu dans un registre .

G. REGISTRES DE BASE.

25 Les registres de base ont le même format 202-203 que les compteurs d'instructions IC et les registres de pile. Les registres de base ^{sont} utilisés au cours du calcul d'adresse pour définir une partie de mémoire. Il y a typiquement huit registres de base de 32 bits respectivement désignés BRO à BR7.

30 H. REGISTRES SCIENTIFIQUES.

Les registres scientifiques (SR) facultatifs sont utilisés pour le calcul sur des nombres binaires en virgule flottante. Il y a typiquement 4 registres scientifiques de 8 octets qui sont désignés SR0-SR3. Les registres scientifiques ont le format 204-205 représenté sur la Fig.2 .

35 I. REGISTRES DIVERS.

Il y a cinq autres registres :

- le compteur d'instructions qui a le format 202-203 ;
- le registre d'état qui a le format 207 ;
- le registre de pile (appelé registre T) ;
- 40 - le registre d'adresse limite (BAR) qui a le format 206 ; et

- le registre de masque de contrôle du matériel qui a le format 208.

Le compteur d'instructions (IC) est un registre de 32 bits qui contient l'adresse de l'instruction en cours d'exécution. Le registre d'état (STR) 207 est un registre de 8 bits qui enregistre des faits-concernant la procédure en cours d'exécution, par exemple si un dépassement de capacité inférieur a été provoqué par l'opération la plus récente. Le registre de pile, appelé également registre T, est un registre de 32 bits qui contient un pointeur désignant le sommet d'une pile associée à la procédure en cours d'activité. Les piles qui seront décrites ci-après, fournissent un espace de travail et un mécanisme pour sauvegarder les variables locales et préserver l'entrée de la procédure et l'information de retour. Le registre d'adresse limite BAR206 est un registre de 28 bits qui spécifie la plus faible adresse absolue de la mémoire qui est accessible au logiciel. Ce registre est chargé lors de l'initialisation du système et peut seulement être lu par le logiciel. Le registre de masque de contrôle du matériel 208 est un registre de 9 bits qui enregistre des informations sur l'état de la machine.

J. FORMATS D'INSTRUCTIONS.

Il y a approximativement 200 instructions, bien que un nombre supérieur ou inférieur puisse être utilisé. Chaque instruction à l'une de quatre longueurs différentes mais elle est toujours constituée par un nombre pair d'octets. Les instructions sont stockées dans des emplacements de mémoire consécutifs. L'adresse de l'octet d'extrême gauche est un multiple de 2 et est l'adresse de l'instruction.

Les huit bits/ plus significatifs (et dans certains cas les bits 8-11 ou 12-15) d'une instruction représentant le code opératoire, tandis que les bits restants représentent un ou plusieurs facteurs. Un facteur peut être un pointeur de registre, un pointeur de déplacement, une syllabe-adresse (adresse logique une valeur littérale), une valeur littérale immédiate. Le type et le nombre des facteurs sont déterminés par le format d'instruction.

III. ORGANISATEUR DU SYSTEME.

A. ETAPE DE TRAVAIL ET TACHE.

Le travail qui doit être exécuté par le système d'ordinateur est défini extérieurement par une série d'étapes de travail, par l'intermédiaire d'un langage de commande de travail. Une étape de travail est une unité de travail à laquelle des ressources du matériel sont attribuées. Typiquement, une étape de travail est constituée de plusieurs tâches. Une tâche est la plus petite unité d'un travail défini par l'utilisateur qui est constitué par un courant d'instructions exécutées sans parallélisme.

B. PROCESSUS.

Les concepts visibles à l'utilisateur de tâche et d'étape de travail

sont respectivement représentés dans le matériel par un processus et un groupe de processus. Un processus est défini comme étant une séquence ordonnées d'instructions qui peut être exécutée de façon asynchrone par l'unité centrale, c'est-à-dire que plusieurs processus peuvent être actifs et partager les ressources mais seul un processus est effectivement en cours d'exécution à un instant donné quelconque. Un groupe de processus est un ensemble associé de processus nécessaires pour l'exécution d'une étape de travail.

C. BLOC DE COMMANDE DE PROCESSUS ET BASE DU SYSTEME.

Du fait que les processus peuvent abandonner la commande de l'unité centrale en divers points au cours de leur exécution, une zone de stockage de la mémoire principale est mise à la disposition d'un processus pour sauvegarder l'état de l'unité centrale. Cette information d'état est utilisée pour préconditionner l'unité centrale avant qu'un processus reprenne la commande de l'unité centrale.

La zone de stockage attribuée à un processus est appelée un bloc de commande de processus (PCB) 400 (Fig. 4). Les données contenues dans un bloc de commande de processus comportent les adresses des zones de mémoire (espaces d'adresses) attribuées au processus, le contenu de tous les registres pertinents et l'état du processus. Ainsi, un bloc PCB sert la zone de stockage temporaire des informations nécessaires pour mettre en route ou relancer un processus, sans perte d'informations. Chaque bloc PCB est accessible par le matériel et peut être adressé par le système d'exploitation au moyen d'un ensemble de tables du matériel microprogrammes établies au cours de l'initialisation du système et modifiées au cours du fonctionnement du système (Fig. 5).

Il existe une zone absolue de la mémoire principale qui est appelée la base du système 302 (Fig. 5 & 6). Cette zone est établie par la microprogrammerie et est accessible par l'intermédiaire du registre d'adresse limite (BAR) 501 qui peut être lu mais non modifié. La base du système 502 contient un certain nombre d'attributs du système parmi lesquels un nom logique (J, P) pour le processus en cours d'exécution. Un autre attribut de la base du système est un pointeur désignant une structure de données définie par le matériel appelée table J. 503. Cette table contient une entrée pour chacune des étapes de travail qui se trouvent dans le système. Chaque entrée de la table J désigne une table P correspondante qui est également une structure de données définie par le matériel. Cette table définit un groupe de processus et contient une entrée pour chaque processus du groupe de processus. Chaque entrée de table P désigne un bloc PCB 400.

Sur la Fig. 5 à laquelle on se référera, le pointeur de la table J 505 indexé par le numéro J, par l'intermédiaire de l'unité de calcul 506, donne

accès à une entrée de la table J 503. Cette entrée contient un pointeur de la table P qui lorsqu'il est indexé par le numéro P par l'intermédiaire de l'unité de calcul 506 donne accès à une entrée de la table P. 504. L'entrée de la table P contient un pointeur 507 désignant le bloc PCB du processus en cours d'exécution. Ainsi, le système d'exploitation peut accéder au bloc PCB en utilisant le contenu du registre BAR 501 et peut accéder à tout autre bloc PCB dont le nom logique (J, P) lui est donné.

D. SEGMENTATION DE LA MEMOIRE

Dans ce cadre de multitraitement, tel que décrit ici, il y a, à tout moment, de nombreux processus contenus dans la mémoire. Ces processus peuvent varier en ce qui concerne leur taille et leurs besoins de mémoire ce qui entraîne des problèmes d'attribution de la mémoire. Le matériel décrit ici, en coopération avec un système d'exploitation (qui n'a pas été représenté ici) résout ce problème par l'attribution dynamique de l'espace de mémoire. Du fait de la nature aléatoire des besoins de mémoire, la mémoire est attribuée en segments de taille variable et l'attribution de la mémoire peut être restructurée au cours du temps de passage de la machine du processus. Ainsi il peut être attribué à un processus un certain nombre de segments de mémoire non contigus. Ce procédé d'attribution de la mémoire est appelée segmentation.

La segmentation présente un problème supplémentaire en ce sens que les adresses en mémoire doivent être modifiées chaque fois qu'une partie ou la totalité d'un processus est translatée. Pour résoudre ce problème le système décrit ici utilise une technique grâce à laquelle les adresses utilisées par un processus sont des adresses logiques et non les adresses absolues en mémoire principale. Les adresses logiques sont utilisées pour développer les adresses absolues.

La segmentation permet également à chaque processus d'avoir accès à ses propres segments de mémoire ou à des segments connexes au moyen d'un système de descripteurs de segments. En accédant à un descripteur de segment, un processus peut obtenir l'adresse d'un segment. Les descripteurs de segments sont contenus dans la mémoire principale et sont entretenus par le système d'exploitation.

Chaque processus peut avoir accès jusqu'à 2068 segments de mémoire normalement, ceci devrait nécessiter un même nombre de descripteurs de segments par processus. Cependant, étant donné que des segments peuvent être partagés, le système d'exploitation groupe les descripteurs de segments en tables de segments. Ce regroupement est basé sur l'accessibilité par un processus (tache), par un groupe de processus (étape de travail) ou une accessibilité globale (ensemble du système). Chaque

processus peut avoir jusqu'à 15 tables de segments coopérant avec lui. Cette technique ne nécessite qu'un descripteur de segment pour chaque segment qui peut être accédé par un processus par l'intermédiaire d'une table de segments. Ainsi l'étape de mémoire nécessaire pour les descripteurs de segment est diminué, la mise à jour de la mémoire au cours de la translation est réduite et une certaine protection du programme est fournie. (Le système de protection principale du programme est le système d'anneaux.

Un processus doit être capable de déterminer quels ont les segments auxquels il a le droit d'accéder. En conséquence, le système donne au processus deux tableaux de mots de tables de segments (STWA). Ces tableaux contiennent les adresses de toutes les tables de segments accessibles à un processus. Il y a deux tableaux de tables de segments par processus, du fait qu'il existe deux tailles de segments, les grands segments et les petits. Les grands segments ont une taille maximale de 2^{22} octets tandis que les petits segments ont une taille maximale de 2^{16} octets. Tous les segments varient de taille par incrément de 16 octets jusqu'au maximum. Un système peut typiquement contenir jusqu'à 28 grands segments et 2040 petits segments.

Les tableaux de mots de tables de segments STWA peuvent être traduits par le système d'exploitation; en conséquence, un processus doit connaître l'adresse absolue de ses tableaux STWA. Le bloc PCB de tout processus contient deux mots qui comportent cette information et qui sont appelés mots d'espaces d'adresses ASW 1 et ASWA2 sur la figure 4. Chaque mot désigne un tableau de mots de tables de segments STWA. Le système d'exploitation met à jour le contenu des mots ASW chaque fois que les tableaux STWA correspondants sont traduits. L'opération consistant à suivre la chaîne de pointeurs et à décoder le descripteur de segment est une fonction de la microprogrammation et ainsi, une fois entreprise n'est pas visible même au système d'exploitation.

La segmentation définit plus de 200 millions d'octets d'espaces d'adresses comme étant disponibles pour les processus. Ce nombre dépasse la capacité de la mémoire principale, en conséquence, une mémoire auxiliaire (à disques ou à tambour magnétiques) est utilisée en combinaison avec la mémoire principale. Le système d'exploitation crée l'illusion que le système a une mémoire principale beaucoup plus grande que celle qui est réellement disponible. Ce concept est appelé "mémoire virtuelle".

A un instant donné quelconque, un segment défini peut être ou ne pas être matériellement dans la mémoire principale. Le contenu d'un descripteur de segment indique si le segment correspondant se trouve ou non dans la mémoire principale. Le matériel détecte toute tentative effectuée par un processus d'accéder à un segment qui ne se trouve pas dans la mé-

moire principale et le notifie au système d'exploitation . Le système d'exploitation provoque le chargement à partir de la mémoire auxiliaire , dans la mémoire principale du segment désiré. Ensuite, le système d'exploitation place l'adresse en mémoire du segment, dans le descripteur de segment qui est le seul endroit où l'adresse absolue d'un segment peut être trouvée. Cette opération est invisible au processus et ainsi il n'est pas averti que le segment ne se trouvait pas dans la mémoire principale ou qu'il peut avoir été translaté dans la mémoire principale.

Le système d'ordinateur décrit ici fournit une protection des données et procédures en empêchant les processus d'interférer entre eux ou de partager leurs espaces d'adresses d'une manière non autorisée . Cette protection est réalisée en restreignant l'adressabilité au moyen de la segmentation de la mémoire et par un système d'anneaux .

Les tables des segments isolent l'espace d'adresses des divers processus contenus dans le système. Les processus utilisent toujours une adresse segmentée au cours de l'exécution . Une adresse segmentée est constituée par un numéro de segment et une adresse relative à l'intérieur du segment .

Le matériel vérifie que l'adresse utilisée par un processus fait partie de l'espace d'adresse attribué au processus. Si l'adresse est en dehors de l'espace d'adresse prescrit , une exception se produit . Un processus ne peut pas faire référence à des données contenues à l'intérieur de l'espace d'adresses d'un autre processus du fait que le matériel utilise les tables de segments du processus qui a émis la référence. Ainsi, il n'y a aucune possibilité qu'un processus ou qu'un groupe de processus puisse référencer une entité appartenant à un autre groupe de processus.

Généralement, un chevauchement d'espace d'adresse dans le système se produit pour ceux des segments qui sont partagés par tous les processus. Ces segments "publics" sont créés par les programmes du système qui effectuent une vérification pour donner une garantie vis à vis des risques de conflits d'adresse . Ainsi, la segmentation protège les programmes d'utilisateurs les uns vis à vis des autres et protège le système d'exploitation vis à vis des programmes d'utilisateurs.

Les segments partagés par plusieurs processus ne sont pas protégés vis à vis des utilisations erronées par l'un de ces processus . Pour résoudre ce problème , un système d'anneaux est utilisé suivant lequel les segments de procédures et de données sont groupées en une hiérarchie à quatre classes. Les quatre classes d'anneaux sont numérotées de 0 à 3.

Chaque anneau représente un niveau de privilège du système, le niveau 0 (anneau le plus intérieur) étant le plus privilégié et le niveau 3

(anneau le plus extérieur) le moins privilégié. A chaque procédure du système sont associés des numéros d'anneau d'exécution minimal et maximal qui spécifient qui peut appeler la procédure. Une procédure est un sous-programme qui est capable d'appeler d'autres procédures et de leur transmettre des paramètres.

Les règles générales du système d'anneaux sont les suivantes :

1. - Procédure dans un anneau intérieur à libre accès aux données dans un anneau extérieur. Inversement, une procédure dans un anneau extérieur ne peut pas avoir accès aux données qui sont dans un anneau intérieur.

2. Une procédure dans un anneau extérieur peut effectuer un branchement sur une procédure dans un anneau intérieur mais l'inverse n'est pas permis.

3. A chaque segment contenant des données, sont attribuées deux valeurs d'anneau, une pour la lecture (RD) et une pour l'écriture (WR).

Ces valeurs d'anneau spécifient la valeur d'anneau maximale dans laquelle une procédure peut exécuter lorsqu'elle accède aux données soit dans le mode d'écriture soit dans le mode de lecture. Chaque fois qu'une instruction de procédure est exécutée, le numéro d'anneau de la procédure (anneau d'adresse effective EAR) est vérifié par rapport aux numéros d'anneau attribués au segment contenant les données référencées. L'anneau d'adresse effective EAR est le numéro maximal des numéros d'anneau contenus dans les registres de base et les descripteurs de données trouvés sur le trajet d'adressage. L'accès aux données est accordé ou refusé sur la base d'une comparaison des numéros d'anneau. Par exemple, si une table du système existe dans un segment ayant une valeur d'anneau de lecture maximale de 3 et une valeur d'anneau d'écriture maximale de 1, une procédure d'utilisateur exécutant dans l'anneau 3 peut lire la table mais ne peut pas la mettre à jour.

Par construction, les anneaux 0 et 1 sont réservés au système d'exploitation et les anneaux 2 et 3 sont réservés à l'utilisateur. L'anneau 0 contient ceux des segments qui sont critiques par le fonctionnement d'ensemble du système. L'anneau 1 contient la masse des segments du système dont la défaillance ne serait pas catastrophique et permettrait un rétablissement. L'utilisateur peut utiliser l'anneau 2 pour les programmes mis au point et l'anneau 3 pour les programmes en cours de mise au point.

F. APPELS DE PROCEDURE.

L'appel de procédure est une fonction importante dans le système décrit ici. Les appels de procédure sont utilisés pour passer d'une procédure à une autre, pour permettre aux procédures d'utilisateur d'utiliser les services du système d'exploitation et pour permettre de réaliser une

structure modulaire à l'intérieur du système d'exploitation. Un appel de procédure est effectué par des instructions et une entité reconnue par le matériel appelée pile (Fig. 7A).

Une pile est un mécanisme qui accepte, met en mémoire et permet l'extraction de données sur la base -dernier-entré -premier-sorti. Les piles résident dans des segments spéciaux appelés segments de pile. Un segment de pile est constitué par un certain nombre de parties contigües appelées éléments de pile 702 (Fig. 7A et 7B) qui sont attribuées dynamiquement à chaque procédure. Le premier élément de pile est chargé au sommet du segment et les éléments suivants sont chargés après lui. Le dernier élément de pile chargé est considéré comme le sommet de la pile. Le registre T702 repère le sommet de la pile pour le processus en cours d'activité. Un registre T virtuel existe dans le bloc PCB de tous les autres processus contenus dans le système.

Un élément de pile 701 représenté sur la Fig. 7B est composé de 3 zones : Une zone de travail 702 servant au stockage des variables, une zone de mémorisation 703 servant à sauvegarder le contenu des registres et une zone de communication 704 servant à la transmission des paramètres entre les procédures. Avant un appel de procédure, l'utilisateur doit spécifier ceux des registres qu'il désire sauvegarder et il doit charger dans la zone de communication les paramètres à transmettre à la procédure appelée. Lorsqu'un appel est effectué, le matériel sauvegarde le contenu du compteur d'instruction IC et des registres de base spécifiés pour faciliter un retour à partir de la procédure appelée.

Chaque appel de procédure crée un élément de pile à l'intérieur d'un segment de pile 701 et les appels imbriqués suivants créent des éléments de pile supplémentaires. Chaque sortie à partir de l'une de ces procédures appelées provoque la suppression d'un élément de pile. Ainsi, un historique des appels est conservé qui facilite un retour ordonné.

Pour assurer une protection entre les procédures exécutant dans les anneaux différents, différents segments de pile sont utilisés. Il y a un segment de pile correspondant à chaque anneau de protection, par processus. Un bloc PCB contient trois mots de base de pile qui désignent le début du segment de pile pour les anneaux 0, 1 et 2 associés au processus. Le segment de pile d'anneau 3 ne peut jamais être entré à partir d'un appel de l'intérieur ; en conséquence, son adresse de départ de pile n'est pas requise dans le bloc PCB.

IV. GESTION ET SYNCHRONISATION DES PROCESSUS.

Le système décrit ici assure des opérations en multitraitement qui sont commandées par un système d'exploitation utilisant une combinaison de

logiciel (ou software), (ou hardware) matériel et micro-programmerie ou firmware. Le logiciel crée et supprime les processus à l'intérieur du système tandis que le matériel et la microprogrammerie assurent la transmission en multiplex des processus à l'unité centrale. En addition, une combinaison de logiciel, matériel et microprogrammerie assure la synchronisation entre processus.

Les processus sont normalement, mais pas toujours, mis en route et arrêtés au commencement et à la fin d'opérations E/S, au cours du traitement de travaux connexes, et d'autres moments, à des fins considérées nécessaires par le système d'exploitation. En conséquence, un système de communication est nécessaire pour mettre en route et arrêter efficacement des processus associés et transmettre les informations entre des processus. Le système de matériel de l'invention comporte des structures de données appelées sémaphores pour établir une voie de communication entre les processus.

A. ETAT DU PROCESSUS.

Un processus peut être dans l'un de quatre états possibles à tout moment ; en cours ; prêt ; en attente ou suspendu. Le matériel reconnaît ces quatre états de processus possibles et exécute diverses procédures de microprogrammerie pour réaliser la distribution des processus, les changements d'état et pour entretenir les structures de données sur la base d'un état du processus. Le bloc PCB contient une zone d'état qui définit l'état actuel de son processus associé.

Un processus est à l'état en cours lorsqu'il a la commande de l'unité centrale. Cet état implique que soit fourni à l'unité centrale un espace d'adresses (tables de segments) et une adresse de départ. L'unité centrale exécute alors les instructions contenues dans les segments de procédures du processus. Le mot de table J du nom du processus (adresse logique) du bloc PCB pour le processus en cours d'exécution est contenu dans le mot de processus en cours (PRW) courant à l'adresse BAR +56 de la base du système (Fig. 6).

L'état prêt est équivalent à l'état en cours, à cette exception près que le processus n'a pas la commande de l'unité centrale (CPU) du fait qu'il n'a pas été reconnu par l'unité centrale. Un processus à l'état prêt est en concurrence pour l'accès à l'unité centrale avec les autres processus prêts et avec le processus en cours.

Un processus est dans l'état d'attente lorsqu'il ne peut se poursuivre jusqu'à ce qu'un événement spécifique se produise, tel que la transmission d'un message par un sémaphore. Un processus en attente n'est pas en concurrence pour l'accès à l'unité centrale, il peut être en concurrence

avec d'autres processus en attente de l'événement requis.

Un processus suspendu est un processus qui a été arrêté pour un certain temps par le logiciel et peut être repris ultérieurement. La décision d'arrêter et de reprendre le processus est extérieure au processus. Ainsi,
 5 un processus suspendu n'est pas actif et ne peut pas en conséquence recevoir la notification de la réalisation d'un événement et ne peut pas utiliser l'unité centrale.

Un processus est suspendu dans les conditions ci-après ;

1. par l'exécution d'une instruction de fin (par suite du fait qu'il a
 10 achevé toutes ses fonctions).
2. par l'exécution d'une instruction de "Suspension" par le système d'exploitation ;
3. par la réalisation d'une condition d'exception par laquelle la commande est transférée au système d'exploitation .

15 B. COMMUTATION DES PROCESSUS.

Les processus passent d'un état à un autre état, soit volontairement par l'action du processus au cours de son exécution, soit involontairement par l'action d'autres processus. Une microprogrammerie de l'unité centrale appelée le Distributeur ou dispatcher commande le mouvement des
 20 processus entre leurs états. Le distributeur utilise une série de files d'attente (qui seront décrites ultérieurement) pour manipuler les processus qui sont dans les états prêts ou en attente. Les processus suspendus sont commandés par le logiciel.

Sur les Fig.8 et 9 auxquelles on se référera , on voit qu'un processus en attente est représenté par un bloc PCB et une entrée de file d'attente spéciale appelée une liaison de processus. La Fig.9 représente une vue éclatée du contenu du segment GO802 et contient les liaisons de processus 803a-803b et 803c-803d des processus actifs et des liaisons de processus libres 805a-805c de processus suspendus. Chaque liaison de processus
 30 spécifie le nom (J, P) du processus, la priorité du processus et , contient un pointeur désignant la liaison de processus suivante dans la file . Il y a divers types de files d'attente, telles que la file d'attente des processus en attente 803a-803b et la file d'attente des processus prêts 803c-803g.

35 Un dispositif du matériel similaire à la table J, appelé table G (Fig.8 et 9) contient des pointeurs désignant tous les segments généraux (connus par l'ensemble des systèmes) 802 à 802n. Le premier élément GO de la table G désigne le segment 802 qui contient les files d'attente du distributeur . Un pointeur de table G801 se trouve dans la base du système
 40 502 (Fig5). De même, dans la base du système se trouve une entrée appelée

mot de file d'attente interne du processeur (Mot IPQW) qui identifie le sommet 803 de la file d'attente des processus prêts 803c-803g dans le segment G0802.

- 5 Ainsi, le distributeur peut examiner tous les processus prêts en consultant la file d'attente des processus prêts 803c-803g. Lorsque le processus en cours d'exécution change d'état, le distributeur enlève la liaison du processus à la tête de la file d'attente des processus prêts et utilise son nom J, P pour accéder à son bloc PCB. Le processus défini par le bloc PCB devient alors le nouveau processus en cours.
- 10 Etant donné que plusieurs processus peuvent être en attente du même événement, une file d'attente de processus en attente 803a-803b existe pour chaque événement. Les processus en attente sont également couplés par l'intermédiaire de liaisons de processus 805 qui résident dans le segment G0. Un pointeur désignant le sommet d'une file d'attente de processus en attente existe dans un sémaphore 903 (qui sera décrit ultérieurement). Il existe un certain nombre d'événements pour lesquels un processus peut être en attente, en conséquence, il y a un certain nombre de files d'attente de processus en attente, à chacune desquelles est associé un sémaphore 903,904.

- Le nombre des processus prêts ou en attente varie de façon dynamique. Ainsi le nombre des liaisons de processus nécessaires pour les files d'attente des processus prêts et en attente varie également. Ce fait introduit un problème de gestion de mémoire pour le distributeur. Le problème est résolu par une autre file d'attente appelée file d'attente des liaisons de processus libres 805a-805c. Cette file d'attente lie entre elles toutes les liaisons de processus contenues dans le segment G0 qui ne sont pas utilisées par les files d'attente des processus prêts ou en attente et peuvent être utilisées pour allonger une file d'attente particulière de processus prêts ou en attente. Un pointeur 901 désignant le sommet 902 de la file d'attente des liaisons de processus libres 805 réside au voisinage du début du segment G0 802.
- 20
25
30

C. SYNCHRONISATION DES PROCESSUS.

- La synchronisation des processus est nécessaire pour coordonner les activités de deux processus travaillant sur la même tâche. La synchronisation est obtenue en utilisant les sémaphores 903-904 qui sont des structures de données résidant dans l'espace d'adresse des processus en communication.
- 35 Un sémaphore est utilisé pour signaler la réalisation d'événements et pour traiter des files d'attente de messages. Un événement, dans ce contexte, est un fait observé par un processus qui peut intéresser un autre processus. Cet événement peut être par exemple /^{l'achèvement} d'une opération asynchrone ou la disponibilité d'une ressource.
- 40

Un processus utilise deux opérations de sémaphore pour signaler la réalisation d'un événement. Une opération d'émission transmet un signal à un sémaphore, l'autre recueille un signal provenant d'un sémaphore (l'opération d'émission est fréquemment appelée opération V ; l'opération de réception est appelée une opération P). L'opération d'émission permet à un processus de transmettre des données ou un signal indiquant que des données sont prêtes. Le sémaphore met le signal en mémoire jusqu'à ce qu'un autre processus soit prêt à le recueillir. Ainsi, le processus émetteur peut librement se poursuivre puisqu'il a transmis les données. L'opération de réception examine un sémaphore spécifié et recueille le signal. Si un signal est présent le processus récepteur continue son exécution. Cependant s'il n'y a pas de signal au sémaphore, le processus récepteur entre dans l'état d'attente. Le sémaphore sert alors de pointeur désignant le sommet d'une file d'attente de processus en attente. Le processus reste dans l'état d'attente dans la file d'attente du sémaphore, jusqu'à ce qu'un autre processus transmette un signal à ce sémaphore particulier. Ainsi, un sémaphore peut conserver un signal jusqu'à ce qu'un processus le recueille ou un sémaphore peut retenir un processus jusqu'à ce qu'un signal lui soit transmis.

Des messages peuvent également être transmis de processus à processus. Un message a la même qualité de présence ou d'absence qu'un signal plus une information supplémentaire. Une partie de l'information est fournie par le matériel et une partie est fournie par la procédure du processus qui transmet le message. Un message porte le nom de processus du processus émetteur. Ainsi, de nombreux processus peuvent transmettre des informations par l'intermédiaire d'un unique sémaphore, les informations étant marquée au nom de l'émetteur.

Un sémaphore à messages peut avoir une file d'attente de messages en attente d'être recueillis par des processus. De même que pour les sémaphores de signaux, les besoins en espace de mémoire s'accroissent et diminuent présentant ainsi un problème de gestion de mémoire. Dans ce cas également, le problème est résolu au moyen d'une file d'attente des liaisons de messages libres. Ces liaisons résident dans un emplacement connu dans un segment qui peut être facilement trouvé lorsqu'il est nécessaire pour fournir ou absorber des liaisons de messages.

Du fait que les sémaphores et les files d'attente établies sur ceux-ci sont partagés par différents processus, l'ensemble de la structure des sémaphores est protégée. Ceci est réalisé par des conventions du matériel et du logiciel qui limitent l'accès à tout segment contenant des sémaphores. Ainsi, les sémaphores doivent se trouver dans des segments de descripteurs

de sémaphore, dont certains peuvent être des segments G (si les communications du système sont nécessaires). Cependant, tous les segments G (sauf le segment GO) sont des segments de descripteurs de sémaphores.

- Chaque descripteur de sémaphore contient un pointeur désignant un sémaphore. Les adresses de sémaphores sont développées par l'Intermédiaire du descripteur de sémaphore ce qui fournit ainsi une protection supplémentaire au sémaphore. Un segment de sémaphore peut être adressé logiquement en utilisant un numéro de segment et un déplacement relatif à l'intérieur du segment ou directement en utilisant les numéros G, D.

10 STRUCTURES DU BLOC DE COMMANDE DE PROCESSUS.

- Sur la figure 4 à laquelle on se référera, on a représenté la disposition du bloc de commande de processus (PCB). Le bloc de commande de processus 400 est une zone de stockage dans la mémoire principale mise à la disposition d'un processus pour sauvegarder l'état de l'unité centrale. L'adressage d'un bloc PCB est effectué comme décrit ci-dessus en se référant à la Fig. 5. Le pointeur de bloc PCB507 (Fig. 5) désigne le bloc de commande de processus PCB à la position d'octet zéro de la Fig. 4. On notera que lorsqu'on progresse vers le bas à partir de l'emplacement 0, les emplacements de mémoire s'accroissent 4 octets par 4 octets tandis que lorsqu'on progresse vers le haut à partir de l'emplacement 0, du PCB ils s'accroissent 8 octets par 8 octets. Les emplacements de mémoire, dans le sens de la progression de haut en bas, sont considérés comme positifs à partir de 0, (à l'intérieur du PCB) tandis que les emplacements de mémoire situés au-dessus de 0 sont considérés comme progressant dans le sens négatif. Certains emplacements situés au-dessus de zéro sont facultatifs et peuvent être ou non inclus dans le bloc de commande de processus (PCB) : de même les emplacements 148 à 176 sont également facultatifs. (On notera que les chiffres placés à côté des emplacements de mémoire spécifient le déplacement des octets à partir de l'emplacement de référence 0 du bloc de commande de processus PCB et ne doivent pas être confondus avec les chiffres de référence couramment utilisés pour identifier des éléments dans un dessin de brevet). En progressant positivement vers le bas, à partir de l'octet 0 jusqu'à l'octet 16 non compris, il y a stockés en mémoire quatre mots principaux du processus PMW0 à PMW3, chaque mot principal de processus PMW ayant une longueur de quatre octets. Le mot principal de processus PMW0 occupe les octets 0 à 3 et est composé de quatre parties : un octet de fonction un octet de priorité, un octet d'état et un octet de décor étendu qui sont représentés sur les fig. 10a à 10d auxquelles on se référera. On a représenté en 10a, les détails du mot principal de processus PMW0, des détails supplémentaires de l'octet de fonction 1001 étant représentés sur la Fig. 10b,

le premier bit 1005 est le bit de mode de comptage servant à indiquer si des fonctions de comptage de temps sont ou non remplies pour le processus. Lorsque le bit de mode de comptage 1005 est mis à la valeur binaire 0, aucune fonction de comptage n'est remplie pour le processus, tandis que lorsque le bit de mode de comptage est mis au 1 binaire, le comptage du temps est exécuté. Le bit de mode scientifique 1006, lorsqu'il est mis à zéro indique que la sauvegarde des registres scientifiques de la machine n'est pas effectuée et que la zone de mémorisation des registres scientifiques situés aux octets 148 à 176 sur la Fig. 4 n'existe pas dans le bloc de commande de processus PCB. Lorsque le bit de mode scientifique 1006 est mis au 1 binaire, l'option scientifique facultative existe et est utilisée dans le processus et la zone de mémorisation des registres scientifiques est utilisée pour sauvegarder le contenu des registres scientifiques lorsque cela est nécessaire. Le bit de mode de reprise 1007 indique lorsqu'il est à 1 qu'un nouvel essai de l'instruction est automatiquement effectué et lorsqu'il est à 0 que ce nouvel essai n'est pas effectué.

Les détails de l'octet de priorité 1002, sont représentés sur la Fig. 10c les quatre premiers bits 1008 de l'octet de priorité sont utilisées pour établir le niveau de priorité du processus correspondant à ce bloc de commande de processus donné PCB. A chaque processus est attribué l'un des 16 niveaux de priorité qui sont utilisées pour ordonner les processus concurrents, c'est-à-dire : (a) pour choisir le processus qui doit être passé en machine parmi les processus prêts, (b) pour mettre les processus dans les files d'attente. Les priorités diminuent de 0 à 15 et, pour un niveau donné de priorité, la règle FIFO (premier - entré premier - sorti) est appliquée. Le bit 4 est le bit d'alarme, le bit 5 n'a pas d'état fonctionnel et les bits 6 et 7 forment le numéro d'anneau d'alarme. L'utilisation des informations contenues dans cet octet sera décrite ultérieurement.

Sur la Fig. 10d à laquelle on se référera, les détails de l'octet d'état 1003 ont été représentés. Un octet d'état est utilisé pour fournir des informations concernant le processus correspondant au bloc de commande de processus PCB400. Le bit d'activité 1010 est mis à l'état binaire 1 lorsque le processus est rendu actif. Le bit de suspension 1011 est mis à l'état binaire 1 lorsque le processus est suspendu. Le champ de sous-état 1012 est un champ de 2 bits et il définit les sous-état ci-après du processus : a) lorsqu'il est mis à l'état binaire 01, le processus est en attente dans la file d'attente des processus prêts (Q/PR/RDY) ; (c) lorsqu'il est mis à l'état 10, le processus est en attente sur un sémaphore dans une file d'attente de sémaphore (Q/PR/S) (d) lorsqu'il est mis à l'état binaire 11, le processus est en cours d'exécution par le processeur. Le champ de mi-

opération MOI1013 est mis à l'état binaire 1 lorsqu'une interruption se produit et est traitée au cours de l'exécution d'une instruction.....c'est-à-dire avant l'achèvement du processus. Le bit de mode de décor étendu

EXTD 1014 est mis à l'état 1 lorsque le processus fonctionne dans un
 5 mode de décor étendu qui peut être un mode d'émulation de la machine. Les bits 1015 et 1016 sont mis à zéro. Le quatrième octet du mot principal de processus PMWO contient le numéro de décor étendu DEXT et est utilisé lorsque le système est dans un mode d'émulation.

Le mot principal de processus PMW1 est stocké dans les octets 4 à 7
 10 du bloc de commande de processus PCB. Les détails du mot PMW1 sont représentés sur la Fig. 10l. L'octet d'état 1016 est le premier octet du mot PMW1 et met en mémoire le contenu du registre d'état. L'octet multiprocesseur MP 1018 est significatif d'une architecture à processeurs multiples : autrement ce champ est égal à zéro. Les second et quatrième octets de mot principal
 15 de processus PMW1 sont respectivement les champs MBZ1017 et 1019 qui doivent être zéro, pour le fonctionnement normal.

Le mot principal de processus PMW2 occupe les octets 8 à 11 du bloc de commande de processus et est représenté en plus de détails sur la Fig. 10f. Sur la Fig. 10f le champ comprenant les bits 4 à 31 contient le nom local
 20 SEG, SRA 1021 du sémaphore auquel le bloc PCB est couplé lorsque le processus est soit à l'état d'attente, soit à l'état suspendu. Le champ de classe et de type d'exception 1023 contient la classe et le type de l'exception du genre interruption qui provoque l'entrée du processus dans l'état suspendu après une exception. Le champ 1022 des bits 4 à 15 est sans utilité lorsque
 25 le processus est dans un état différent de ceux mentionnés ci-dessus.

Le mot principal de processus PMW3 occupe les octets 12 à 15 du bloc PCB400 et désigne une table d'extension de décor. Sur la Fig. 10g, à laquelle on se référera, les détails du mot PMW3 ont été représentés, le champ DETSZ (taille de la table du décor étendu) 1024 définit le nombre d'en-
 30 trées de la table et si ce champ est égal à 0, aucune extension de décor n'est permise au processus. Le champ DETA (adresse absolue de la table de décor étendu) 1025 est l'adresse absolue de la table de décor étendu en unités de 16 octets et n'est significatif que si le champ DETSZ n'est pas nul. La table de décor étendu est formée par les entrées de table DETSZ.
 35 Chaque entrée a une longueur d'un octet. La DEXT^{ème} entrée de la table définit la capacité du processus à fonctionner dans le mode de décor étendu DEXT. Lorsque le DEXT^{ème} octet est 0 le numéro de décor étendu DEXT n'est pas accordé tandis que le DEXT^{ème} octet est 1 le numéro de décor étendu est accordé. Des valeurs du numéro DEXT autres que 0 et 1 sont
 40 incorrectes (cf. Fig. 10a numéro DEXT 1004).

Les octets 16 à 23 du bloc PCB 400 contiennent deux mots d'espace d'adresses, respectivement appelés ASWO et ASW1, et chaque mot ASW contient un pointeur désignant un tableau de mots de tables de segments. Les mots ASWO et ASW1 ont tous deux le même format représenté sur la

5 Fig. 10h. La taille du tableau de mots de tables de segments est définie par le nombre de mots de tables de segments contenus dans un tableau et comprend typiquement 6 mots pour le mot ASWO et huit pour le mot ASW1. Le champ STWSZ 1026 indique la taille du tableau de mots de tables de segments. Le champ de tableau de mots de tables de segments STWA 1027 contient l'adresse absolue STWA du tableau en unités de 16 octets c'est-à-dire

10 que l'adresse absolue du tableau est égale à 16 fois la valeur STWA en octets.

Les octets 24 à 27 du bloc PCB contiennent un mot d'exception EXW représenté sur la Fig. 10i en plus de détail. Le mot d'exception contient un

15 pointeur (SEG, SRA) 1029 désignant une table de classes d'exceptions qui, définit l'action qui doit être entreprise à la suite d'une exception de processus selon sa classe telle qu'elle est mise en mémoire dans le mot principal de processus PMW2 (cf Fig. 10f). Le champ MBZ 1028 du mot EXW doit être zéro.

20 Le mot de pile SKW qui occupe les octets 28 à 31 du bloc PCB contient la valeur du sommet du registre T de la pile du processus lorsque le processus n'est pas en cours d'exécution et est représenté avec plus de détails sur la Fig. 10j. Sur la Fig. 10j les bits 0 et 1 définissent le champ TAG 1030. Le champ TAG 1030 indique le type de descripteur par son contenu et

25 doit être zéro pour le mot de pile SKW. Les bits 2 et 3 du mot SKW contiennent le champ RING 1031 qui contient le numéro d'anneau associé à l'adresse segmentée de la pile aux fins de protection et dans ce cas, ils doivent être zéro. Les bits 4 à 31 contiennent le numéro de segment SEG et l'adresse relative dans le segment SRA et constituent un champ qui identifie le

30 segment décrit dans une table de segments et l'adresse relative SRA à l'intérieur du segment. Le mot de pile SKW est mis à jour chaque fois que le processus quitte l'état "en cours" ou courant. Il est utilisé pour restaurer le contenu du registre T chaque fois que le processus passe en exécution. Dans ce dernier cas, les champs TAG 1030 et RING 1031 sont vérifiés pour

35 contrôler qu'ils sont des zéros ; autrement, une exception de bloc PCB incorrect se produit.

Les octets 32 à 35 du bloc PCB 400 contiennent le mot contenu dans le compteur d'instruction ICW qui peut être appelé également ICC. Sur la Fig. 10k à laquelle on se référera, on a représenté les détails du mot du compteur

40 d'instruction ICW dans lequel le champ TAG 1033 doit contenir la valeur

blinaire 00 (c'est-à-dire des valeurs autres que zéro sont incorrectes dans le compteur d'instructions). Le champ d'anneaux courant RING COURANT 1034 qui occupe mes positions de bits 2 et 3 définit le numéro courant du processus qui doit être utilisé par la détermination des droits d'accès à la mémoire principale. Les bits 4 à 31 contiennent le numéro de segment et l'adresse relative dans le segment (SEG, SRA 1035) qui définissent l'adresse de l'instruction suivante à exécuter.

Le champ MBZ des octets 36 à 39 doit être zéro.

(On notera que le champ MBZ désigne toujours un champ qui doit être à zéro). Le mot MBZ est vérifié chaque fois que le bloc PCB est appelé à partir du nom J, P. S'il n'est pas égal à zéro, une exception PCB incorrect se produit.

Les mots de base de pile SBWO à SBW2 occupent les octets 40 à 51 du bloc de commande de processus 400. Ces mots ont tous le même format qui est représenté en plus de détails sur la Fig. 101. Ils sont utilisés au cours d'opérations de pile et chaque fois qu'ils sont utilisés, leur champ TAG 1036 et leur champ RING 1037 doivent être zéro, autrement, une exception de PCB incorrect se produit. Les bits 4 à 31 contiennent l'adresse segmentée (SEG, SRA) 1038 des premiers octets des segments de pile respectifs des anneaux 0, 1 et 2.

Les octets 52 à 83 du bloc de commande des processus 400 sont un espace réservé à la zone de mémorisation des registres de base (8 mots). Les octets 84 à 147 constituent une zone de mémorisation qui est utilisée pour sauvegarder les valeurs de tous les registres généraux (16 mots). Les octets 148 à 176 constituent une zone de mémorisation qui est utilisée pour sauvegarder les registres scientifiques (8 mots). Cette zone étant optionnelle.

Cinq mots doubles sont prévus dans le bloc PCB400 au-dessus de l'adresse zéro du bloc PCB, aux fins de comptage de temps lorsque le bit de mode de comptage du mot PCB est mis à l'état 1. Ces mots sont situés de l'adresse PCB-8 à l'adresse PCB-40. Chaque mot contient un temps ou un intervalle de temps exprimé en microsecondes dans ses 52 premiers bits, les bits 52 à 63 étant remplis de zéro. Le mot double du temps d'exécution partiel RTO (les 8 premiers octets négatifs au-dessus de zéro dans le bloc PCB) contient la durée qui sera effectivement dépensée par le processeur pour le compte du processus avant qu'une exception de temps expiré se produise. Le mot RTO est mis à jour de la façon suivante : chaque fois que le processus sort de l'état en cours, la valeur du compteur de temps du processus est stockée dans le mot RTO. Chaque fois que le processeur entre dans l'état en cours, la valeur du compteur de temps

du processus est chargée à partir du mot RTO.

Le mot double de comptage d'exécution totale RUA occupant les octets 9 - 16 est un compteur de temps qui spécifie la durée totale du temps du processeur pendant lequel un processus a été dans l'état en cours. Le temps
5 compté est le temps effectivement dépensé par le processeur exclusivement pour le compte du processus. Le mot ^{RUA} est mis à jour de la manière suivante :

Chaque fois que le processus sort de l'état en cours, la valeur du compteur de temps de processus PT est lue. La différence entre le contenu du mot RTO et celui du compteur PT est ajoutée au mot RUA (la valeur PT est
10 ajoutée, consécutivement, au mot RTO). On notera que le temps pendant lequel le processus est suspendu n'est pas calculé. Les mots RTO et RUA sont mis à jour même si le bit de mode de comptage est mis à 0. Cependant, les mots CET, RTA et WTA (qui seront décrits ci-après) ne sont prévus dans le bloc de commande de processus que si le bit de mode de comptage
15 du mot principal de processus PMWO est mis à l'état 1. Ils ne sont mis à jour que dans ce cas.

Le mot de comptage de temps d'attente WTA aux octets 17 à 24 est un compteur de temps réel qui spécifie la durée de temps réel totale pendant laquelle le processus a été à l'état d'attente. Le mot WTA est mis à
20 jour de la manière suivante ; chaque fois que le processus sort de l'état d'attente la valeur de l'horloge donnant l'heure du jour (non représentée) TOD est lue et la valeur de l'heure du jour TOD, moins la valeur du mot CET, est ajoutée au mot WTA.

Le mot de comptage du temps prêt RTA qui occupe les octets 25 à 32
25 est un mot double qui est un compteur de temps réel qui spécifie la durée totale du temps réel pendant laquelle le processus a été à l'état prêt. Le mot RTA est mis à jour de la manière suivante : chaque fois que le processus quitte l'état prêt, la valeur de l'heure du jour TOD est lue et le contenu TOD moins le contenu du mot CET est ajouté au mot RTA.

30 Le mot double de l'heure de début courante CET qui occupe les octets 33 à 40 contient l'heure du jour à laquelle le processus est entré dans l'un des états ci-après ; prêt, en attente, en cours, suspendu.

STRUCTURES DE LA BASE DU SYSTEME.

Sur la Fig. 6 à laquelle on se référera la configuration de la base
35 du système 600 a été représentée. La base du système réside dans la mémoire principale, est engendrée par la microprogrammerie et est accessible par l'intermédiaire du registre d'adresse limite (BAR), qui peut être lu mais non modifié. Le registre d'adresse limite BAR est situé au-dessous d'une zone de la mémoire principale réservée au matériel et sépare cette zone
40 de la mémoire réservée au matériel de la base du système 600. Sur la Fig. 6

à laquelle on se référera maintenant, la base du système 600 contient un certain nombre d'attributs du système parmi lesquels un numéro d'étape de travail et un numéro de groupe de processus (J,P) pour le processus en cours d'exécution. A partir du nom logique J, P du processus, l'adresse absolue du bloc de commande de processus PCB correspondant est obtenue. La taille est l'adresse de la table J sont définies par le contenu du mot de table J (JTW). Ce mot est situé à l'adresse définie par le registre BAR. Le format du mot JTW est représenté sur la Fig. 11a. La taille JTSZ 1101 de la table J1204 (Fig. 12) définit le nombre d'entrées ou éléments de la table J 1204 qui peut contenir jusqu'à 256 entrées. Le champ JTSZ 1101 est un nombre entier positif de 8 bits ou une exception "hors de la table J" se produit si le numéro J est supérieur à la valeur JTSZ. L'adresse absolue de la table J1204 est obtenue en multipliant par 16 le pointeur de table J1102. La table J1204 contient les entrées de table J dont le format est représenté de façon plus détaillée sur la Fig. 11b. Chaque entrée de la table J définit l'adresse absolue d'une table P 1205 qui est obtenue en multipliant par 16 le pointeur de table P. 1104. La taille (PTSZ) 1103 d'une table P définit le nombre d'entrées de la table P. Le champ PTSZ est un nombre positif de 8 bits qui peut typiquement être compris entre 0 et 255 pour indiquer le nombre d'entrées de la table P. Une exception "hors de table P" se produit si le numéro P est supérieur à PTSZ. Chaque entrée de la table P 1205 définit l'adresse absolue d'un bloc de commande de processus (PCB) 1206 qui est obtenue en multipliant par 16 le pointeur de bloc PCB 1107. Un indicateur de présence P1105 indique l'absence d'un bloc PCB 1206 lorsqu'il est à l'état binaire 0 et indique la présence du bloc PCB lorsqu'il est mis à l'état binaire 1. (Lorsque l'indicateur de présence P1105 est trouvé à l'état 0, une exception d'entrée de table P vacante se produit). Les bits 1 à 7 de l'indicateur de table P (Fig11) 1106 doivent être 0 (MBZ), autrement il se produit une exception d'entrée de table P incorrecte.

A l'adresse BAR+4 de la base du système 600 se trouve l'octet de taille du mot de table G (GTW) représenté en plus de détails sur la Fig. 11d. La taille d'une table de segments G1212 (Fig. 12) est définie par le contenu d'un mot de table G (GTW). La taille (GTSZ) 1108 de la table G1212 définit le nombre d'entrée de la table G qui peut typiquement s'élever jusqu'à 256 entrées. Le champ GTSZ est un nombre entier positif de 8 bits. Une exception "hors de la table G" se produit si le numéro G est supérieur à la taille GTSZ. L'adresse absolue d'une table G1212 est obtenue en multipliant par 16 le pointeur de table G 1109. Le format de l'entrée de segments G a une longueur de deux mots (8 octets) et est appelé descripteur de

segment G. Le format de descripteur de segment G est représenté en détail sur les Fig. 11e et 11f. Tous les descripteurs de segments G sont directs et en conséquence le bit "indirect" I, 1111 doit être 0, autrement il se produit une exception de descripteur de segment incorrect. L'Indicateur de présence

5 1110 est un champ d'un bit qui, lorsqu'il est mis à l'état binaire 1, indique qu'un segment est défini dans la mémoire principale pour le numéro de segment auquel le descripteur correspond, tandis que s'il est remis à zéro, aucun segment n'est défini et une référence au descripteur de segment provoque une exception de segment manquant. Le bit de disponibilité

10 A1112 est un champ d'un bit qui indique si le segment est ou non disponible; il n'est vérifié que si ce segment est défini (c'est-à-dire si P est égal à 1 binaire), autrement il est ignoré. Le champ indicateur d'utilisation U1113 indique si le segment a ou non été accédé. Si le bit U est mis à la valeur binaire 0, le segment n'a pas été accédé tandis que si le bit U est mis à la

15 valeur binaire 1, le segment a été accédé. Le bit Indicateur d'écriture W1114 indique si une opération d'écriture a été effectuée dans le segment. Si ce bit W est mis à l'état binaire 0, aucune écriture n'a été effectuée dans le segment tandis que si le bit W est à l'état 1 une opération d'écriture a été effectuée dans le segment. L'Indicateur de commutation GS1115 d'un des-

20 cripteur de segment G doit être mis à la valeur binaire 01, autrement il se produit une exception de descripteur de segment incorrect. La raison de ce fait est qu'un segment G contient toujours des sémaphores (bien que l'inverse ne soit pas nécessairement vérifié; c'est-à-dire que tous les sémaphores ne doivent pas se trouver nécessairement dans un segment G) et les instruc-

25 tions sur sémaphores exigent que le code GS soit à la valeur binaire 01. L'adresse absolue de la base du segment 1214 est définie dans le descripteur de segment G de la Fig. 11e par le champ de base de 24 bits 1116; le contenu de ce champ est multiplié par 16 pour obtenir l'adresse absolue. Le second mot du descripteur de segment G représenté sur la Fig. 11f occupe

30 les positions binaires 32 à 63 dans la table G1212. Le champ RSU 1117 qui occupe les bits 32 à 39 est réservé pour l'utilisation du logiciel et est généralement ignoré lorsqu'utilisé comme descripteur de segment G comme il l'est dans le présent cas. Le champ MBZ 1118 doit être 0; autrement il se produit une exception de segment incorrect. Etant donné que le champ MBZ 1118

35 occupe les bits 40 à 51 il détermine la longueur du champ TAILLE 1119 qui est le champ pour un petit segment. En conséquence, tous les segments G doivent être des segments du type petit segment. La taille 1119 du segment est un nombre entier positif de 12 bits définissant le nombre d'octets du segment et la taille du segment est interprétée comme multiple de 16. En

40 conséquence, la taille du segment d'un segment G1214 ne peut dépasser 2^{16}

octets (petits segments).

On se référera à nouveau à la base du système 600 représentée sur la Fig. 6. Il y a 9 mots de cellule d'exception du système situés entre les adresses BAR + 8 et BAR + 44. Le format des mots de cellule d'exception du système (EXC) est représenté sur la Fig. 11g. Etant donné que des sémaphores sont utilisés pour transmettre des messages aux processus consacrés lorsqu'une exception du système se produit, les pointeurs désignant ces sémaphores sont trouvés dans 9 emplacements de mémoire, chaque emplacement étant appelé une cellule d'exception du système, une par classe d'exception du système. Le champ MBZ 1120 doit être mis à la valeur binaire 0, autrement un contrôle du système se produit. Chaque cellule d'exception contient le nom de système G 1121 D1122 du sémaphore.

La cellule d'exception du canal située à l'adresse BAR+44 de la base du système 600 a un format qui est similaire à celui de la cellule d'exception du système précédemment décrit et contient le nom de système GD d'un sémaphore qui est utilisé pour transmettre des messages à des processus déterminés lorsqu'une exception de canal se produit.

Un mot de file d'attente interne du processeur IPQW est situé à l'adresse commençant à BAR + 48 et les détails de son format sont représentés sur la Fig. 11h. Le mot IPQW désigne le sommet d'une file d'attente de processus prêts (Q.FR.RDY) telle que celles représentées sur la Fig. 9 et désignées par les références 905 et 805. La file d'attente des processus prêts (Q.FR.RDY) lie tous les processus qui sont à l'état prêt. Elle est référencée par le champ de tête de file Q.FR.RDY 1124 (Fig. 11h) du mot IPQW qui indique le sommet de la file d'attente des processus prêts. Le champ de tête de file Q.FR.RDY 1124 contient un nombre entier de 16 bits qui est le déplacement par rapport à la base du segment G N°0, appelé segment GO, du premier octet de la file d'attente Q.FR.RDY. Si ce champ Q.FR.RDY est zéro, la file d'attente des processus prêts est considérée comme étant vide. Le champ MBZ 1123 doit être égal à zéro s'il en est autrement un contrôle du système se produit.

A l'adresse BAR + 52 de la base du système 600, on a représenté les positions de stockage pour le comptage de relances initial et pour le comptage de relances courant dont le format est représenté en détail sur la Fig. 11i. Le champ NFS 1125 est un champ de stockage non fonctionnel et n'est pas utilisé par la base du système. Le champ de comptage de relances initial 1126 et le champ de comptage de relances courant sont utilisés pour déterminer le nombre de fois où une relance d'instruction automatique est exécutée avant qu'une erreur machine soit produite pour créer une condition d'exception de défaillance machine. Ils sont chargés avec le même nombre par une

instruction de restauration de comptage de relances (non représentés ici).

- Le mot de processus en cours (RPW) représenté sur la Fig. 11j est mis en mémoire à l'adresse BAR + 56 de la base du système⁶⁰⁰ et est utilisé pour garder en mémoire le nom du processus en cours avec sa priorité dans le cas d'une architecture comportant un processeur unique. Les champs NFS 1128 et 1131, respectivement, sont des champs de mémoire non fonctionnels et peuvent être utilisés à des fins quelconques par des moyens quelconques de système mais ils ne sont en général pas utilisés par la base du système. Le niveau de priorité d'un processus en cours est stocké dans le champ PRI 1129.
- Un bit de déroutement asynchrone ou bit d'alarme est mis en mémoire dans le champ AB 1130 tandis qu'un anneau de déroutement asynchrone est mis en mémoire dans le champ ARN 1132. Le nom logique J, P du processus en cours dans le cas d'une architecture à un seul processeur est mis en mémoire dans le champ J, P 1133.
- Un mot pointeur de table d'absolutisation représenté sur la Fig. 11k est situé à l'adresse BAR + 60 de la base du système et est utilisé lors du chargement initial du système pour initialiser les adresses absolues dans le programme de chargement initial du système (ISL) en ajoutant le contenu du registre BAR à toutes les adresses absolues du programme ISL. Le pointeur d'absolutisation 1135 définit l'emplacement d'une table d'absolutisation (non représentée). La taille de la table d'absolutisation est indiquée par le champ ATSZ 1134. Le mot de numéro de série du Processeur représenté sur la Fig. 11l est un mot de quatre octets situé à l'adresse BAR + 64 et contient le numéro de série du processeur dans le champ de numéro de série du processeur 1136.
- Un mot de limite supérieure de la mémoire principale représenté sur la Fig. 11m est situé à l'adresse BAR + 68 et indique dans le champ 1139 la limite supérieure de la mémoire principale en donnant l'adresse absolue du dernier mot disponible dans la mémoire principale.
- A l'adresse BAR + 72 se trouve un mot, représenté sur la Fig. 11n pour indiquer le numéro de canal (CN) de l'unité de chargement initial du système ISL (champ 1140) et le numéro de canal (CN) du dispositif matériel (champ 1141).
- Le type et le sous-type d'un dispositif utilisé dans le système d'ordinateur sont représentés, dans un mot de type de dispositif matériel (Fig. 11q) respectivement par les champs 1143, et 1144, le champ RSU 1142 de ce mot étant réservé au logiciel. Ce mot se trouve dans la base du système à l'adresse BAR + 76. Un mot similaire ayant un format type similaire représenté sur la Fig. 11p contient le type et le sous-type du dispositif utilisé pour le chargement initial. Ce mot se trouve à l'adresse BAR + 80.

Lorsqu'on appuie sur le bouton de "reprise" d'un ordinateur une opération V simulée est exécutée sur un sémaphore et l'état "prêt" est entré. Un pointeur indiquant ce sémaphore se trouve à l'adresse BAR +84 de la base du système 600 et est appelé mot de cellule de relance. Ce mot a le format

5 représenté sur la Fig. 11q. Le format est similaire à celui du mot de cellule d'exception du système décrit ci-dessus et contient le nom de système G, D d'un sémaphore respectivement dans le champ G1149 et le champ D1150. Le champ MBZ 1148 doit avoir la valeur 0.

Dans le cas où il y a plusieurs processeurs raccordés au système

10 d'ordinateur, un mot est prévu dans la base du système à l'adresse BAR+88 pour l'extension multiprocesseur, les détails de ce mot sont représentés sur la Fig. 11.

EXEMPLES D'UTILISATIONS DE LA BASE DU SYSTEME ET DU BLOC DE COMMANDE DE PROCESSUS.

15 Sur la Fig. 12 à laquelle on se référera, on a représenté un exemple de la manière suivant laquelle la base du système peut être utilisée en combinaison avec le bloc de commande de processus PCB afin d'adresser et d'accéder à un segment d'utilisateur, à un segment du système ou à un segment de file d'attente de processus prêts (Q/PR/RDY). La mémoire

20 principale 1200 comporte une partie 1203 dont l'utilisation est réservée au matériel. Un registre d'adresse limite BAR 1202 sépare la base du système 1215 de la partie de la mémoire 1203 réservée au matériel. Le registre d'adresse limite BAR 1202 est utilisé pour adresser des éléments de la base du système en ajoutant le contenu du registre BAR au déplacement, en unités

25 de 4 octets, de l'élément désiré de la base du système. Cette adresse indique alors le premier octet de l'élément désiré de la base du système, sur la Fig. 12, le registre BAR 1202 désigne le mot de table J (JTW). Le mot de table J comme précédemment indiqué, comporte un pointeur qui désigne une table J1204. En indexant le pointeur avec le numéro J représenté sur la

30 Fig. 5, une entrée de table J 1216 est obtenue. Dans l'entrée de table J se trouve un pointeur de table P qui indique l'adresse absolue de la table P1205. En indiquant cette adresse avec le numéro P (Cf Fig. 5) dans la table P1205 l'adresse absolue du bloc de commande de processus 1206 est obtenue. Comme précédemment décrit, il existe dans le bloc de commande de

35 processus PCB 1206, deux mots d'espaces d'adresses ASWO et ASW1. Les bits de gauche du champ de numéro de table des segments STN du registre de base 1201 sont utilisés pour accéder à l'un de ces deux mots d'espace d'adresses, dans l'exemple représenté le mot ASW1, qui comporte un pointeur de tableau de mots de tables de segments STWA qui désigne le ta-

40 bleau des mots de tables de segments STWA 1208. En combinaison avec le

numéro de table de segments STN du registre de base 1201, l'un des huit mots de table de segments est accédé dans le tableau STWA 1208 qui désigne l'une des 8 tables de segments 1210, l'entrée de table de segments STE du registre de base 1201 est alors utilisée pour sélectionner l'une des 256 entrées de la ta-
 5 ble de segments 1210 où se trouve un descripteur de segment. Le descripteur de segment est alors utilisé pour accéder à un segment d'utilisateur 1211.
 (Pour plus de détails on se rapportera à la demande de brevet N°7317765 du 16 Mai 1973 ayant pour titre Dispositif de développement et de calcul d'adresse dans une mémoire segmentée au nom de la demanderesse.

10 Afin d'accéder à un segment du système 1214 qui est utilisé pour stocker les sémaphores, un mot de table G est utilisé dans la base du système 1215. L'adresse du mot de table G est obtenue en ajoutant le déplacement du mot de table G dans la base du système au registre d'adresse limite BAR 1202 (Cf Fig. 6). Le mot de table G, GTW comporte un pointeur de table G
 15 qui désigne une table G1212. En utilisant un numéro G que le système peut obtenir et en indexant la table G à l'aide de ce numéro, un descripteur de segment G est accédé qui est utilisé pour adresser un segment du système 1214.

20 D'une manière similaire, la base du système 1215 est utilisée pour accéder à la file d'attente des processus prêts (Q/PR/RDY) 1213 en recherchant un mot de file d'attente interne du processeur IPQW qui désigne le segment Q/PR/RDY 1213.

UNITE DE COMMANDE

25 Sur les Fig. 13a à 13c auxquelles on se référera on a représenté les détails de l'unité de commande. L'unité de commande, bien que représentée séparée de l'unité centrale (CPU) est en fait une partie de l'unité centrale et est constituée d'une unité de mémoire de commande (CSU) 1301, d'un adaptateur d'interface de la mémoire de commande CIA 1302 et des sous-unités coopérantes, à savoir le chargeur de mémoire de commande CSL 1303 et
 30 l'unité de chargement et de contrôle CLU 1304.

L'unité de mémoire de commande CSU 1301 reçoit des micro-instructions du chargeur de mémoire de commande CSL 1303 par l'intermédiaire de l'unité de chargement et de contrôle CLU 1304 et de l'adaptateur d'interface CIA 1302. Dans les conditions de fonctionnement normales, des microprogrammes sont chargés à partir d'une source externe au cours de l'initialisation du système et deviennent une fonction de commande permanente de la machine. Cependant, l'unité de mémoire de commande CSU 1301 a la capacité d'être rechargée et initialisée d'une manière qui permet une diversité de modes opérationnels de l'unité centrale CPU 1306. Les modes de
 35 fonctionnement suivants de l'unité centrale sont disponibles, sous la com-
 40

mande de l'unité de mémoire de commande CSU 1301 : (a) le mode natif ;
 (b) le mode d'émulation ; (c) le mode natif et d'émulation simultanés ; (d) le
 mode de diagnostic . Cette capacité est rendue possible du fait que les mi-
 cro-instructions résidants dans la mémoire de commande CSU sont la source
 5 de micro-opérations utilisées pour commander le fonctionnement de toutes les
 autres unités fonctionnelles de l'unité centrale CPU, telles que l'unité
 d'émulation 1316, l'unité arithmétique et logique ALU 1317, l'unité de poin-
 tage d'instruction 1318 IFU, l'unité de contrôle d'adresse ACU 1319 , et
 l'unité de gestion des données DMU 1321. On a également représenté , com-
 10 me étant contenus dans l'unité centrale CPU1306 , les registres généraux
 1307, les registres de base 1308 , les registres scientifiques 1309, le re-
 gistre T 1310 , le registre d'état 1311 , le compteur d'instruction IC 1312
 et le registre de masque de contrôle du matériel 1313, précédemment décrits.

Typiquement, l'unité de mémoire de commande CSU 1301 est une mé-
 15 moire morte programmable à circuits intégrés bipolaires de 9k (9216 octets)
 (FROM mélangée à une mémoire à accès sélectif à écriture /lecture (RAM).
 Cette unité CSU a typiquement un cycle de lecture de 150 nanosecondes
 et un cycle d'écriture de 450 nanosecondes . Chaque emplacement de la mé-
 moire de commande met en mémoire un mot de microinstruction de 84 bits
 20 (qui sera décrit plus en détail ci-après), et chaque mot de microinstruction
 commande un cycle de l'unité centrale CPU. Lorsque chaque emplacement
 de mémoire de commande de l'unité de mémoire de commande CSU1301 est
 lu, son contenu est décodé par des décodeurs de microopérations qui four-
 nissent des signaux de commande de microopérations ; chacun de ces si-
 25 gnaux provoquant l'exécution d'une opération spécifique (qui sera décrite
 ultérieurement en détail) à l'intérieur de l'unité centrale CPU.

En groupant des positions à l'intérieur de chaque mot de microinstruc-
 tion (qui sera décrit en détail ultérieurement) de séquences de mémoire de
 commande sont obtenues qui peuvent exécuter une opération ou instruction
 30 spécifique de l'unité centrale. Lorsque chaque instruction est déclenchée
 par l'unité centrale, certains bits contenus dans le code opération sont uti-
 lisés pour déterminer la séquence de départ de la mémoire de commande. Le
 contrôle de certaines bascules (non représentées qui sont enclenchées et
 restaurées par les fonctions de décodage d'instructions permet à la mémoi-
 35 re de commande d'effectuer un branchement sur une séquence plus spéci-
 fique lorsque cela est nécessaire.

L'adaptateur d'interface de la mémoire de commande CIA 1302 commu-
 nique avec l'unité de mémoire de commande 1301 , l'unité de gestion des don-
 nées DMU 1321 , l'unité de contrôle d'adresse ACU 1319 et l'unité arithméti-
 40 que et logique ALU 1317 pour diriger le fonctionnement de la mémoire de

commande 1333, représentée sur la Fig. 13b2. L'adaptateur d'interface CIA 1302 comporte des circuits logiques pour la modification, la vérification, le contrôle d'erreur, des adresses de la mémoire de commande et pour la génération d'adresses du matériel. La génération d'adresse du matériel est utilisée en général, pour le développement de l'adresse de départ des séquences d'erreur ou pour la séquence d'initialisation.

L'unité de gestion des données DMU 1321 constitue l'interface entre l'unité centrale 1306 et la mémoire principale et/ou la mémoire tampon représentées sur la Fig. 1. C'est la responsabilité de l'unité de gestion des données de reconnaître quelle est l'unité qui contient les informations requises par d'autres unités et de permettre le transfert de ces informations aux registres appropriés de l'unité centrale, au temps approprié. L'unité de gestion des données DMU effectue également le masquage au cours des opérations d'écriture partielle.

L'unité de pointage IFU 1318 communique avec l'unité de gestion des données DMU 1321, l'unité de contrôle d'adresse ACU 1319, l'unité arithmétique et logique ALU 1317 et l'unité de mémoire de commande CSU 1301 et elle a la responsabilité de mettre en temps utile des instructions à la disposition de l'unité centrale. L'unité de pointage d'instruction IFU 1318 comporte l'instruction suivante disponible dans ses registres avant l'achèvement d'instruction en cours. Pour obtenir cette capacité, l'unité de pointage d'instruction IFU 1318 contient un registre d'instructions de 12 octets (non représenté) qui contient normalement plusieurs instructions. En addition, l'unité IFU 1318, sous la commande de l'unité de mémoire de commande CSU 1301, requiert les informations (instructions) de la mémoire principale avant que l'instruction soit effectivement devenue nécessaires maintenant ainsi son registre d'instructions de 12 octets constamment à jour. Les instructions sont ainsi préalablement extraites au moyen de cycles mémoire normalement inutilisés. L'unité de pointage d'instruction décode également chaque instruction et informe les autres unités de la longueur et du format de l'instruction.

L'unité de contrôle d'adresse ACU 1319 communique avec l'unité de pointage d'instruction IFU 1318, l'unité arithmétique et logique ALU 1317 l'unité de gestion de données DMU 1321 et l'unité de mémoire de commande CSU 1301 par l'intermédiaire de l'adaptateur d'interface CIA 1302. L'unité de contrôle d'adresse ACU 1319 est responsable de tous les développements d'adresse dans l'unité centrale. Toutes les opérations de l'unité de contrôle d'adresse, y compris les transferts à l'unité, à partir de l'unité et à l'intérieur de l'unité, sont dirigés par les micro-opérations de l'unité de mémoire de commande et par les circuits logiques de l'unité ACU. Le cycle

normal de l'unité de contrôle d'adresse ACU dépend des types d'adresses contenues dans l'instruction plutôt que du type de l'instruction. Selon les types d'adresse, l'unité de contrôle d'adresse ACU peut effectuer différentes opérations pour chaque adresse contenue dans une instruction. L'unité

5 de contrôle d'adresse ACU contient également une mémoire associative 1319a qui, typiquement, met en mémoire l'adresse de base des 8 segments de mémoire les plus récemment utilisés avec leurs numéros de segment. Chaque fois qu'une demande d'accès à la mémoire est effectuée, le numéro de segment est comparé au contenu de la mémoire associative pour déterminer si l'a-

10 dresse de base du segment a déjà été développée et mise en mémoire. Si l'adresse de base est contenue dans la mémoire associative 1319a, cette adresse est utilisée dans le développement de l'adresse absolue et un temps considérable est économisé. Si l'adresse de base n'est pas contenue dans la mémoire associative 1319a, elle est développée en accédant aux tables de

15 mémoire principale. Cependant, après que l'adresse de base de segment a été développée, elle est stockée dans la mémoire associative avec le numéro de segment, en vue de futures références.

L'unité arithmétique et logique ALU 1317 comporte une interface avec l'unité de contrôle d'adresse ACU 1319, l'unité de pointage d'instruction IFU 1318, l'unité de gestion des données DMU 1321 et l'unité de mémoire

20 de commande CSU 1301. L'unité ALU 1317 a pour fonction première d'exécuter les opérations arithmétiques et les manipulations des données requises de l'unité centrale. Les opérations de l'unité arithmétique et logique sont complètement dépendantes de signaux de commande de micro-opérations

25 provenant de l'unité de mémoire de commande CSU 1301.

Une mémoire de travail et mémoire bloc-notes LSU 1315 (appelée quelquefois également mémoire locale) coopère avec l'unité arithmétique et logique ALU 1317 et la mémoire de commande CSU 1301. Cette mémoire de travail est typiquement constituée par une mémoire transistorisée à 256

30 emplacements (32 bits par emplacements) et par des circuits logiques de sélection et de lecture/écriture pour cette mémoire. La mémoire bloc-notes 1315 est utilisée pour stocker les informations de commande de l'unité centrale et les informations sur les facilités d'entretien. En outre, la mémoire bloc-notes 1315 contient des emplacements de travail qui sont principalement

35 utilisés pour la mise en mémoire temporaire des facteurs et des résultats partiels au cours de la manipulation des données. Est également associée à l'unité arithmétique et logique ALU 1317, une mémoire auxiliaire 1317a constituée typiquement de 64 bascules bistables servant à conserver en mémoire des états divers du système d'ordinateur.

40 L'unité centrale comporte également une unité de synchronisation

1320 qui constitue essentiellement deux systèmes d'horloge en un. Le premier système d'horloge engendre les impulsions de synchronisation pour l'adaptateur d'interface CIA 1302 et le second système d'horloge engendre les impulsions de synchronisation pour les opérations des unités fonctionnelles à l'intérieur de l'unité centrale.

Sur la Fig. 13c à laquelle on se référera on a représenté le format du mot de mémoire de commande 1325. Le mot de mémoire de commande a typiquement 84 bits et est divisé en 6 zones ou "champs" principaux.

- a) type de séquence 1326 (3 bits) ;
- b) branchement et/ou microopérations 1327 (23bits) ;
- c) génération et désignation des constantes 1328 (14bits) ;
- d) données vers le bus 1329 (8bits) ;
- e) microopérations 1330 (32 bits) ;
- f) contrôle 1331 (4bits) ;

Le champ E de 3 bits du mot de mémoire de commande 1325 est utilisé comme champ de commande de séquence. Il y a typiquement 7 types de séquences différents et 1 type réservé pour le présent système d'ordinateur. Comme indiqué dans le bloc 1335 de la Fig. 13b1, lorsque le champ E est égal à la valeur binaire 0, 1 ou 2, le champ de branchement A, B, C, D et L de la microinstruction 1325 est utilisé pour engendrer l'adresse suivante. Les six premiers bits du registre d'adresse KS 1337 sont utilisés en combinaison avec le champ B, les résultats de test C, les résultats de test D et le champ L, pour fournir l'adresse suivante de la microinstruction suivante qui est alors placée dans le registre d'adresse KS 1337. Lorsque le champ E est mis à la valeur binaire 4 (cf bloc 1335 Fig. 13b1), l'adresse suivante sélectionnée est extraite du registre de retour d'interruption KA 1339. L'adresse conservée dans le registre KA est celle engendrée par la logique de génération de l'adresse suivante lorsqu'une interruption du matériel se produit. Lorsque le champ E est mis à la valeur binaire 5, un branchement est utilisé pour déclencher un sous-retour à partir d'un sous-programme du microprogramme. Lorsque le champ est utilisé, le contenu du registre de retour KR 1346 est utilisé en tant qu'adresse suivante de la mémoire de commande. Le registre de retour 1346 est chargé par l'émission d'une commande de la mémoire de commande qui charge l'adresse présente de la mémoire de commande contenue dans le registre KS 1337, plus 1 de l'incrémenteur 1338 dans le registre KR 1336. Une capacité de sous-programme à imbrication d'un niveau supplémentaire est effectuée par l'intermédiaire du registre de branchement de retour KT 1347. Chaque fois que le registre KR 1346 est chargé, l'ancien contenu du registre KR est transféré au registre KT 1347 ; chaque fois que le retour du microprogramme

est appelé, le contenu du registre KT est transféré au registre KR. Une capacité de sous-programme avec une imbrication à trois niveaux est fournie par le registre KU 13 48 et une capacité de sous-programme avec imbrication à quatre niveaux est offerte par le registre de branchement de retour KV1349.

- 5 Lorsque le champ E du mot de mémoire de commande est ^{mis} à la valeur binaire 6, le mot suivant de la présente mémoire de commande qui est adressé est égal à l'adresse présente du registre KS 1337 plus 1 ajouté par l'incrémenteur 1338. Lorsque le champ E est mis à la valeur binaire 7, la mémoire de commande CSU entre dans le code de diagnostic et l'adresse suivante est l'adresse présente plus 1.

- 10 En plus de la commande de séquence de branchement à l'adresse suivante de la mémoire de commande décrite ci-dessus est représentée dans le bloc 1335, il existe une commande de séquence engendrée par le matériel représentée dans le bloc 1336 de la Fig. 13b1 (à noter : les blocs 1335 et
- 15 1336 sont en pratique des registres matériels qui ont été ainsi représentés pour montrer les différentes formes que les mots de microinstruction peuvent prendre). Les branchements engendrés par le matériel sont des conditions prioritaires (telles qu'erreurs, initialisations, balayage de la mémoire de commande, etc.....) qui annulent le champ E et introduisent une adresse
- 20 fixe dans le registre d'adresse KS 1337 de la mémoire de commande. Le branchement est effectué en portant une ligne d'interruption (non représentée à un haut niveau pendant une période d'horloge et en stockant l'adresse qui aurait été engendrée sous la commande du champ E dans le registre de retour d'interruption KA 1339. Une adresse engendrée par le matériel est
- 25 alors placée dans le registre d'adresse de la mémoire de commande. Certaines interruptions engendrées par le matériel/microprogrammerie ont priorité lorsque est mise la bascule (non représentée) de blocage d'interruption qui empêche que des interruptions supplémentaires dans leur classe puissent être exécutées jusqu'à ce que la condition d'interruption en cours ait été satisfaite.
- 30 Il existe une microopération de la microprogrammerie pour commander la restauration de la bascule de blocage d'interruption pour celles des séquences qui sont sous la commande de la microprogrammerie. Les séquences qui sont sous la commande du matériel engendre automatiquement une restauration de la bascule de blocage à la fin de la séquence. Les conditions ci-après
- 35 énumérées par ordre de priorité, existent dans cette catégorie : a) chargement de la mémoire de commande b) balayage de la mémoire de commande c) erreur matériel d) erreur logiciel. Les autres conditions du matériel n'enclenchent pas la bascule de blocage d'interruption mais provoquent l'exécution d'une action immédiate lorsqu'elles sont engendrées. Les conditions
- 40 suivantes, énumérées par ordre de priorité existent dans cette catégorie :

- a) initialisation ;
- b) effacement logiciel (soft clear)
- c) entrée panneau maintenance ;
- d) entrée canal maintenance ;
- e) sortie matériel ;

5

Un signal d'initialisation amène la mémoire de commande CSU1301 à se brancher à l'adresse binaire 0, à effacer les erreurs restaurables par le matériel et à exécuter une opération de chargement de la mémoire de commande suivie de la séquence de balayage de la mémoire de commande sous la commande du matériel. Ce signal provoque également l'initialisation du système. Un signal d'effacement du logiciel amène la mémoire de commande CSU 1301 à se brancher à l'adresse binaire 0, à effacer les erreurs restaurables par le matériel et à restaurer la bascule de blocage d'interruption. Un signal d'entrée du panneau de maintenance amène la mémoire de commande CSU 1301 à se brancher à l'adresse préétablie dans les commutateurs d'adresse de la mémoire de commande sur le panneau de maintenance (non représenté).

15

Un signal d'entrée de canal de maintenance amène la mémoire de commande CSU 1301 à se brancher à une adresse engendrée par l'intermédiaire du canal de maintenance (non représentée). L'adresse chargée provient du bus de maintenance QMB 1344 qui fait partie du canal de maintenance et est justifiée à droite. Un signal de sortie de matériel provoque le branchement de la mémoire de commande CSU à l'adresse binaire 2. Cette séquence est utilisée comme moyen de maintenance. A la fin de la séquence, un retour est déclenché en émettant un branchement par le champ E, avec le champ E mis à la valeur binaire 4.

20

25

Un signal de chargement de la mémoire de commande amène la mémoire de commande à se brancher à l'adresse binaire 0. Ce signal ferme également la bascule, non représentée, du cycle de lecture de la mémoire de commande CSU, arrête l'horloge du système 1320 et met la mémoire de commande CSU dans l'état de chargement. Dans l'état de chargement la mémoire de commande peut être chargée à partir du chargeur de mémoire de commande CSL 1303, du contrôleur d'entrée/sortie IOC 1305, de la mémoire principale 102 ou du panneau de maintenance 1355. Lorsqu'elle est chargée à partir du chargeur CSL 1303, un balayage automatique est engendré à la fin du chargement. Lorsqu'elle est chargée à partir d'un autre moyen quelqu'il soit, un balayage peut être commandé soit en engendrant un signal de microopération soit en enclenchant le commutateur de balayage sur le panneau de maintenance. Un signal de balayage de la mémoire de commande provoque le branchement de la mémoire CSU à l'adresse binaire 0. Un

30

35

40

balayage de la mémoire de commande est sous la commande du matériel pendant toute la durée de la séquence. Au cours du balayage, l'horloge du système 1320 est arrêtée et en conséquence aucune commande ou test n'est exécuté. A la fin de la séquence de balayage, le matériel transfère le contenu du registre de retour d'interruption KA1339 au registre d'adresse KS 1337, l'horloge du système est remise en route et la commande est rendue à la microprogrammation.

Un signal d'erreur du matériel provoque le branchement de la mémoire de commande CSU à l'adresse binaire 4. Dans le mode de traitement normal une erreur du matériel détectée dans une quelconque unité fonctionnelle de l'unité centrale CPU rend active une ligne d'erreur du matériel (non représentée). La séquence de mémoire de commande engendrée teste et vérifie les conditions du système pour déterminer l'action à entreprendre. Dans le mode de diagnostic, les conditions d'erreur qui peuvent être détectées par le matériel sont visibles aux microdiagnostics. Les microdiagnostics commandent l'action à entreprendre. Un signal d'erreur du logiciel d'autre part, provoque le branchement de la mémoire de commande à l'adresse binaire 1. Cette adresse est le départ de la séquence de compte-rendu d'erreur du logiciel qui est sous la commande de micro-programmes.

Sur la Fig13c à laquelle on se référera à nouveau, le champ E 1326 est un champ de 3 bits pour le code de branchement, comme précédemment décrit. Le champ de branchement et/ou de microopérations 1327 est composé des champs A, B, C, D, et L (représentés également dans le bloc 1335 de la Fig. 13b1) dans lesquels le champ A représente les six bits de gauche de l'adresse suivante. Le champ B, les 4 bits centraux de l'adresse suivante du champ masque sur un branchement à 64 voies, le champ C est un champ de test de 6 bits pour l'un de 64 tests, le champ D est un autre champ de test de 6 bits pour l'un des 64 tests et le champ L est le bit le moins significatif. Le champ K 1328 est un champ de 14 bits dont six bits sont destinés aux champs de constante, 4 bits sont destinés à un champ de constante ou un champ d'orientation et 4 bits constituent un champ d'orientation pour une constante. Le champ "données vers bus" 1329 est composé du champ QA ayant 4 bits pour commander les informations transmises à la partie QA du bus QMB 1344 et le champ QB qui comporte 4 bits pour commander les informations transmises à la partie QB du bus QBM 1344. Le champ F1330 est un champ de 32 bits qui est codé pour engendrer des sous-commandes de microopérations. Le champ P1331 est composé de 4 bits réservés au contrôle.

En fonctionnement, les mots de microinstruction sont stockés dans le tableau de la mémoire de commande 1333. Au cours d'un cycle de

fonctionnement , la mémoire de commande est adressée par le contenu du registre d'adresse KS 1337. Ceci provoque la lecture du contenu de l'emplacement spécifié par l'adresse dans le groupe de bascules de lecture 1357. Des parties du contenu du mot qui se trouve dans les bascules de lecture sont

5 réparties ou transférées à des registres de stockage à l'intérieur de chacune des unités fonctionnelles de l'unité centrale. Chaque unité fonctionnelle comporte des circuits logiques de décodage pour engendrer les sous-commandes requises spécifiées par le mot de mémoire de commande sous la commande de l'horloge du système. En général, le décodage est effectué à l'intérieur de

10 chacune des unités fonctionnelles de l'unité centrale plutôt que d'être effectué centralement afin de réduire au minimum le temps de décodage et de réduire le nombre des cables qui seraient normalement nécessaires pour transmettre les signaux de commande si le décodage était effectué centralement. En outre, le décodage est effectué à l'intérieur de chaque unité pour

15 éviter les problèmes de synchronisation provenant des différences des retards de transmission des cables. Au surplus, en décodant les sous-commandes à l'intérieur de chaque unité, ceux des signaux qui sont représentatifs de certaines conditions existant à l'intérieur de l'unité fonctionnelle et sont nécessaires pour la génération de certains signaux de sous-commande n'ont

20 pas été retournés à l'adaptateur CIA 1302. Un décodeur typique 1359 a été représenté sur la Fig. 13b2 comme recevant divers champs de mots de microinstructions et engendrant des signaux de microopération A, B, C, D.... Q, R. Un décodeur de microopérations typiques 1359 reçoit des commandes d'un mot de microinstruction. Le champ du mot de microinstruction est décodé et porte à un haut niveau l'une d'une pluralité de lignes

25 s, t, u.....y, z. Une matrice est formée en ayant une impédance de ligne de commande prédéterminée couplée aux lignes s-z aux points $\alpha, \beta, \gamma, \dots, \psi, \omega$. Typiquement, alors, lorsque le champ d'une microinstruction est décodé, l'une des lignes s-z passe à un haut niveau. Etant donné que les cercles désignés sur la matrice par les lettres grecques α, β, γ re-

30 présentant le couplage inactif entre les deux séries de lignes, tout signal électrique se propageant sur un fil horizontal est couplé de façon à se propager sur un fil vertical A à R lorsqu'un couplage inductif (cercle) est indiqué. Chaque ligne verticale A - R peut alors ^{être} couplée pour former l'une

35 des entrées d'une porte ET respective 1360 à 1365. D'autres signaux d'entrée peuvent être appliqués aux portes ET 1360 à 1365 et notamment un signal de synchronisation TS provenant de l'unité de synchronisation centrale. En conséquence, lorsque chaque signal TS passe à un haut niveau, celles des portes dont toutes les autres entrées sont à un haut niveau sont

40 rendues passantes et fournissent des signaux de microinstruction à des unités

fonctionnelles prédéterminées de l'unité centrale. Par exemple, si une commande 1341 des bascules de lecture 1357 est décodée et que la ligne horizontale est portée à un ^{haut} niveau, les lignes de commande verticales A, B, C et Q sont portées à un haut niveau et les portes ET 1360, 1361, 1362 et 1364 sont rendues actives, lorsque le signal de synchronisation TS est appliqué à ces portes, séquentiellement. En conséquence, la combinaison suivant laquelle les lignes de commande verticales sont couplées à la ligne de commande horizontale à différents points représentés par les lettres grecques α ω ψ représente une matrice de commutation permanente pour fournir des signaux de microopérations à l'unité centrale CPU afin de commander les unités fonctionnelles à l'intérieur de l'unité centrale par des microinstructions fournies par la mémoire de commande 1333. Ainsi une microprogrammerie permanente présentant la caractéristique de pouvoir être modifiée peut être incorporée dans la machine de cette invention simplement en spécifiant la séquence de microopérations qui est requise en tant que l'une des capacités du système d'ordinateur.

Dans les conditions normales, les données sont écrites dans le tableau de mémoire de commande 1333 par l'intermédiaire du registre des données d'écriture de l'unité centrale appelé également registre local YO 1343. Une bascule de commande (non représentée) définit si l'écriture doit être effectuée dans la moitié supérieure ou dans la moitié inférieure du tableau de mémoire. Les données provenant de l'unité de chargement et de contrôle CLU 1304 arrivent dans l'adaptateur d'interface et dans l'unité de mémoire de commande CIA/CSU par l'intermédiaire du bus de maintenance QMB 1344 et sont mises en mémoire intermédiaire dans le registre local de stockage YO 1343 avant d'être écrites dans la mémoire de commande 1333. Le registre de stockage local YO 1343 est utilisé en temps partagé à la fois comme registre d'écriture et comme registre de lecture local. Le multiplexeur 1345 peut être commandé soit par le panneau de maintenance 1355 soit par des microdiagnostics et établit un trajet de lecture en sortie à partir des registres qui y sont connectés. Un registre de comparaison KP 1350 est prévu pour une utilisation non fonctionnelle et est utilisée essentiellement aux fins de la maintenance et est utilisée en combinaison avec la logique Egalité KS = KL 1352 et la logique de décodage 1351.

35 MICROPROGRAMMERIE DU DISTRIBUTEUR POUR LA COMMANDE DES PROCESSUS.

Le distributeur est une unité de microprogrammerie /matériel dont l'objet principal est de gérer les diverses files d'attente de processus et d'effectuer l'aiguillage entre les processus et il comporte les fonctions de mise à jour de s files d'attente de processus, des blocs de commande de pro-

cessus (PCB) du mot de processus en cours dans la base du système et des
 registres d'un nouveau processus. Il délivre également un message à un
 processus qui est effectivement en attente d'un tel message sur un sémaphore
 (après une opération V, une opération V simulée pour le contrôle d'entrée-sor-
 5 tie IOC, pour le manipulateur d'exceptions). Il met également un message dans
 la file d'attente d'un sémaphore après une opération P qui libère une liaison
 de message lorsqu'un processus est en attente sur le sémaphore de liaisons
 libres pour délivrer son message.

L'unité distributeur appelle en outre la microprogrammerie d'in-
 10 structions, du mode natif après "rappel" d'un processus exécuté dans le mode
 natif ou après un conflit si le processus courant reste en cours et est exé-
 cuté dans le mode natif. Elle appelle également la microprogrammerie d'ex-
 tension de décor pour :

- a) appel temporaire au cours du "retrait" d'un processus exé-
 15 cuté dans le décor étendu ;
- b) un appel temporaire au cours du "rappel" d'un processus exé-
 cuté dans ce décor étendu ;
- c) un appel définitif à la fin du "rappel" d'un processus exécuté
 dans ce décor étendu ;
- 20 d) un appel définitif après "conflit" si le processus courant reste
 en cours et est exécuté dans ce décor étendu.

En outre, le distributeur place le système dans la boucle de mar-
 che à vide lorsqu'il n'y a pas de processus en cours.

Il existe plusieurs manières d'entrer ou de sortir d'un distribu-
 25 teur qui sont les suivantes :

1. La procédure d'initialisation SIP fournit une entrée à sa
 dernière étape (cf demande de brevet N° 73 42704 du 30 Novembre 1973
 ayant pour titre "Mécanisme de comptage hardware/firmware pour un sys-
 tème informatique" au nom de la demanderesse.
- 30 2. L'instruction de départ et de suspension assure une entrée
 dans le distributeur. L'instruction "Départ" démarre un processus, l'in-
 struction "suspension" termine un processus (voir demande de brevet N°
 73 42674 du 30 Novembre 1973 ayant pour titre "Système de gestion de pro-
 cessus pour un processeur central" au nom de la demanderesse.
- 35 3. Les opérations P & V assurent une entrée dans le distributeur.
 L'opération P capte un message provenant d'un sémaphore et s'il n'y a
 pas de message, le processus se met alors à l'état d'attente (cf demande de
 brevet n° 73 42691 du 30 Novembre 1973 ayant pour titre "Synchronisation de
 processus par utilisation de sémaphores" au nom de la demanderesse.

40 En résumé, en conséquence le distributeur est le principal mécanisme

qui gère les processus, et en conséquence les blocs de commande de processus PCB, en décidant quel est le processus qui doit être passé en machine puis en exécutant l'action appropriée, telle que le retrait du processus en cours d'exécution (à savoir : écrire dans le bloc PCB toutes les informations relatives au processus en cours d'exécution contenues dans les registres du matériel, la mémoire de travail etc.....) et le rappel du nouveau processus (écrire, en les extrayant du bloc PCB dans les divers registres du matériel, la mémoire de travail etc..... toutes les informations nécessaires pour le passage machine d'un nouveau processus).

Les fonctions remplies par le distributeur sont représentées sur les organigrammes des Fig. 14a à 14i. Par exemple, le bloc 1402 de la Fig. 14a1 représente la fonction remplie par le distributeur suivant laquelle un mot de microprogramme est fourni par l'unité de mémoire de commande 1301 et, à la suite de son décodage par le décodeur 1359, ce mot commande aux parties intéressées de l'unité centrale par l'intermédiaire d'une série appropriée de signaux de microopérations 1360, 1361 etc... d'extraire le mot IPQW (mot de file d'attente interne du processeur) de le transférer à la mémoire de travail ou mémoire bloc-notes 1315. Simultanément le distributeur extrait, à l'étape 1404, le descripteur de segment GO (cf Fig 12) de la table des descripteurs de segments désignée par le mot de table GTW dans la base du système. Les bits 16-31 du mot IPQW contiennent un nombre entier positif de 16 bits qui est le déplacement à partir de la base du segment G numéro 0, appelé segment GO, de la tête (premier octet) de la file d'attente des processus prêts (Q/PR/RDY). Si les bits 16-31 du mot IPQW sont à 0, comme déterminé à l'étape 1403, la file d'attente des processus prêts est considérée comme vide. Si la file d'attente des processus prêts est vide cela indique qu'il n'y a pas de processus actuellement en attente dans la file Q/PR/RDY et la file d'attente des processus prêts est vide. La question suivante qui doit être décidée (étape de décision 1405) est de déterminer s'il y a ou non un processus en cours d'exécution dans la machine, en déterminant si l'indicateur LIBRE est à 1. L'indicateur LIBRE est une bascule située dans la mémoire auxiliaire 1317a, il est mis lorsqu'il n'y a pas de processus courant CJP en cours d'exécution dans le processeur. Si l'indicateur LIBRE est mis (c'est-à-dire qu'il n'y a pas de processus en cours d'exécution), et étant donné qu'il a été précédemment déterminé qu'il n'y a pas de processus, dans la file d'attente des processus prêts à utiliser le processeur, la machine se met alors à l'état de repos ou à vide (à l'étape 1406). Cependant, s'il y a un processus actuellement en cours d'exécution dans la machine, mais qu'il n'y a pas de processus en attente

d'utiliser la machine le processus en cours d'exécution accède à son instruction suivante, à l'étape 1407. En retournant maintenant à l'étape de décision 1403 de l'organigramme de la Fig. 14a1, si la zone de pointeur du mot IPQW (à savoir les bits 16 à 31) contient un nombre entier positif, la tête de file d'attente des processus prêts désignée par le mot IPQW dans le segment GO, est extrait et chargée dans la mémoire de travail (mémoire bloc-notes) (N.B pour éviter des répétitions et pour plus de clarté de la description, les fonctions intermédiaires exécutées par le distributeur, en combinaison avec l'unité de mémoire de commande et l'unité central e ont été omises ci-après ; cependant il est bien entendu que de telles fonctions intermédiaires, telles que celles précédemment décrites à titre d'exemple, sont typiquement présentes). A ce point, il a été déterminé qu'il existe un processus en attente dans la file d'attente des processus prêts. Avant qu'une autre action puisse être entreprise, il est nécessaire de déterminer s'il y a un processus actuellement en cours d'exécution dans le processeur central, ceci est déterminé à l'étape de décision 1410, de l'organigramme et s'il n'y a pas de processus en cours d'exécution, le processus en tête de la file d'attente des processus prêts est mis en exécution, à l'étape 1412. Cependant, si il y a un processus en cours d'exécution dans le processeur central, le distributeur doit déterminer celui des deux qui a priorité : le processus en cours d'exécution ou celui en tête de la file d'attente des processus prêts. En conséquence, l'octet de priorité du processus courant (CJP) qui est situé dans le mot de processus en cours de la base du système 600, ou dans le mot principal de processus PMWO du bloc PCB 400 est extrait à l'étape 1413. Une décision est alors prise à l'étape 1414, pour déterminer si le processus en cours d'exécution CJP a une priorité inférieure à celle du nouveau processus NJP en attente en tête de la file d'attente des processus prêts (étape de décision 1414). Si le processus en cours CJP n'a pas une priorité inférieure à celle du nouveau processus NJP, le processus CJP conserve la commande du processeur central et l'indicateur de conflit est remis à 0 à l'étape 1415a. L'indicateur de conflit est toujours mis à zéro sauf lorsqu'un ou plusieurs nouveaux processus ont été placés dans la file d'attente des processus prêts depuis le début de la dernière instruction exécutée pour le compte du processus en cours CJP créant ainsi la possibilité d'un conflit, dans ces conditions l'indicateur de conflit est mis à la valeur binaire 1. Avant que le processus courant CJP ait l'autorisation de se poursuivre et d'exécuter de nouvelles instructions, cependant un contrôle est effectué pour déterminer si le processus courant CJP est exécuté dans un mode de décor étendu, à l'étape 1415. Si le processus CJP est exécuté dans le mode de décor étendu

l'instruction suivante est exécutée dans le mode d'émulation (c'est-à-dire dans le mode de décor étendu (étape 1419a) et si le processus n'est pas exécuté dans le mode de décor étendu, l'instruction suivante est exécutée dans le mode natif (étape 1419b). Revenant à nouveau à l'étape de décision 1414,

5 si le nouveau processus NJP en tête de file d'attente des processus prêts a une priorité supérieure à celle du processus courant CJP (c'est-à-dire que son numéro de priorité est inférieur au numéro de priorité du processus CJP, le processus actuellement en cours CJP est retiré de la machine et le nouveau processus NJP est rappelé dans la machine. En conséquence, un

10 sous programme de priorité de la microprogrammerie PRIQ, à l'étape 1418, commande la mise en file d'attente du processus courant CJP, dans la file d'attente des processus prêts suivant l'ordre dernier entré, premier sorti et le numéro de priorité en réalisant tout d'abord le retrait du processus courant CJP dans la direction du sous-programme de retrait RLLO de la microprogrammerie (1ère étape 1419). Le sous-programme de retrait RLLO

15 dirige l'écriture des informations du processus CJP qui sont stockées dans les registres généraux, les registres de base, les registres scientifiques dans le registre T, le registre d'état et le compteur d'instruction, en retour/des zones de mémorisation appropriées du bloc de commande de processus PCB et il commande la mise à jour du mot de comptage d'exécution totale RUA. En

20 outre le numéro du mot principal de processus PMWO du bloc PCB 400 est mis à jour à l'étape 1420.

Le nouveau processus NJP est alors prêt à être appelé. Le registre d'adresse limite BAR est extrait à l'étape 1422 et le mot de processus en cours RPW est extrait à l'adresse BAR+56, à l'étape 1423. Le nom du nouveau processus NJP est ensuite écrit dans le mot de processus en cours RPW et étant donné que le nom du nouveau processus NJP était écrit dans la liaison de processus PL de la file d'attente Q/PR/RDY, le nom contenu dans la liaison de processus PL est en conséquence placé maintenant dans

30 le mot RPW, à l'étape 1424. En conséquence, nouveau processus NJP extrait de la file d'attente des processus prêts devient maintenant le processus en cours CJP et a le droit de commander le processeur central et, en conséquence, il n'est plus en attente dans la file d'attente Q/PR/RDY et doit être retiré de la file d'attente par le retrait

35 de son nom hors de la liaison de processus PL de la file d'attente Q/PR/RDY à l'étape 1425. Lorsque ceci est effectué la file d'attente des processus prêts Q/PR/RDY est mise à jour par le sous-programme de la microprogrammerie UQLK à l'étape 1425a. En conséquence, le numéro JP du processus qui a juste été sorti de la machine est placé, à l'étape 1426 dans une liaison de

40 processus dans la file d'attente Q/PR/RDY étant donné qu'il n'a plus la

commande de la machine et doit attendre pour l'obtenir. A ce point la transformation consistant à donner la commande du processeur central au nouveau processus et à placer l'ancien processus dans une file d'attente des processus prêts est achevée et du fait qu'il y a un processus (le nouveau processus en cours CJP) qui a la commande du processeur central, l'indicateur LIBRE est mis à zéro à l'étape 1427. Si par contre, aucun processus CJP n'avait la commande du processeur central l'inducteur libre serait mis à l'état 1. A ce point, l'attribution du processeur est terminée et un nouveau processus a acquis la commande du processeur central tandis que l'ancien processus a été placé dans une file d'attente des processus prêts. Cependant, le nouveau processus n'est pas encore prêt à être exécuté car le matériel de l'unité centrale 1306 (Fig. 13a1) tel que les registres généraux 1307, les registres de base 1308, les registres scientifiques 1309, le registre T 1310, le registre d'état 1311 et le compteur d'instruction 1312 doivent recevoir les informations de commande du bloc de commande de processus du nouveau processus.

En conséquence, le sous-programme de micro-programmerie extrait tout d'abord le mot PMW3 du bloc PCB (Fig. 4) et le charge dans la mémoire de travail ou mémoire bloc-notes 1315 (étape 1432) puis il extrait le mot PMWO (étape 1431). Le champ MBZ du mot PMWO est contrôlé, à l'étape 1433, et s'il n'est pas égal à 0, une exception de bloc-PCB - incorrect se produit. Cependant, si le champ MBZ du mot PMW O est 0, le mot PMW1 est extrait, à l'étape 1434. Egalement, le champ MBZ du mot PMW1 est contrôlé pour déterminer s'il a ou non la valeur binaire 0. Si ce champ n'est pas un 0 binaire, il se produit une exception de bloc PCB incorrect, tandis que s'il est égal à 0 le distributeur passe à l'étape C.

En conséquence, le mot d'espace d'adresses ASWO est extrait de l'emplacement approprié du bloc PCB (étape 1436) et le champ de taille du mot de table des segments STWSZ est contrôlé, à l'étape 1437, pour déterminer s'il est supérieur à 7 ou inférieur. S'il est supérieur à 7 il en résulte une exception de bloc PCB incorrect, s'il est inférieur ou égal à 7, le mot ASW1 est extrait à l'étape 1438, et son champ STWSZ est contrôlé à l'étape 1439 pour déterminer s'il est ou non inférieur ou égal à 8. Si ce champ est supérieur à 8, une exception de bloc PCB incorrect se produit. Cependant, si le champ STWSZ est égal ou inférieur à 8, le mot d'exception EXW est extrait à l'étape 1440, et son champ MBZ est testé à l'étape 1441 pour déterminer s'il est ou non égal à zéro ; si son champ MBZ n'est pas égal à 0, une exception de bloc PCB incorrect se produit ; tandis que s'il est égal à 0, le mot de pile SKW est extrait à l'étape 1442 et son champ MBZ est testé à l'étape 1443 pour déterminer s'il est ou non égal à zéro. Si le champ

MBZ n'est pas égal à 0, il se produit une exception de bloc PCB incorrect tandis que s'il est égal à 0, le mot de compteur d'instruction ICW est extrait du bloc PCB et placé dans le compteur d'instruction IC (à l'étape 1444) et son champ TAG est testé pour déterminer s'il est ou non égal à 0, à l'étape 1445. Si le champ TAG du mot ICW n'est pas égal à 0, il se produit une exception de bloc PCB incorrect. Cependant, si le champ TAG est égal à 0, le mot MBZ est extrait à l'étape 1446 et son champ MBZ (bits 0-31) est testé pour déterminer s'il est ou non égal à 0, à l'étape 1447. S'il n'est pas égal à 0, il en résulte une exception de bloc PCB incorrect, tandis que s'il est égal à 0, les mots de base de pile 0, 1 et 2 SBWO, 1, 2 sont extraits à l'étape 1448. Le contenu des 8 registres de base contenus dans la zone de mémorisation des registres de base du bloc PCB est alors extrait à l'étape 1449, et mis en mémoire dans les registres de base 1308 de la machine. Ensuite le contenu des 16 registres généraux de la zone de mémorisation des registres généraux du bloc PCB est extrait à l'étape 1450, et stocké dans les registres généraux 1307 de la machine. Avant d'extraire le contenu des registres scientifiques cependant un contrôle de l'octet de fonction du mot principal de processus PMWO est effectué pour déterminer si le mode scientifique est ou non utilisé à l'étape 1451. Si le mode scientifique est utilisé, le contenu des registres scientifiques de la zone de mémorisation des registres scientifiques du bloc PCB est extrait et stocké à l'étape 1452. La microprogrammerie entreprend alors l'exécution d'une analyse de l'octet de fonction du mot PMWO pour déterminer si le mode de comptage est utilisé (étape 1453). Si le mode de comptage est utilisé (c'est-à-dire si le bit de comptage d'octet de fonction est mis à la valeur binaire 1) les mots de comptage existent dans le bloc PCB et le mot de comptage de temps prêt RTA est mis à jour. Ensuite, la microprogrammerie entreprend de déterminer si le numéro DEXT est ou non mis à 0 (étape 1454). S'il n'est pas à 0, il indique que la machine peut être dans le mode émulé (c'est-à-dire que la capacité d'extension de décor est utilisée) et, en conséquence le numéro DEXT du mot PMWO est vérifié, à l'étape 1445 pour déterminer s'il est supérieur ou inférieur au champ DETSZ du mot PMW3 et s'il est supérieur une exception de bloc PCB incorrect se produit, si le numéro DEXT est inférieur au champ DETSZ mais n'est pas égal à 0, la machine fonctionne dans un mode émulé autorisé et l'on passe à l'étape F. Retournant à l'étape de décision 1454, si le champ DEXT a la valeur binaire 0, le mode natif est exécuté par la machine et la machine extrait les mots de pile STW à l'étape 1457. Le mot de temps d'exécution partiel RTO du bloc PCB est extrait à l'étape 1458 et le compteur de temps du processus est chargé avec la limite de temps que le processus CJP

peut passer à l'état en cours.

A ce point soit (a) un nouveau processus NJP a été rappelé pour prendre la commande de l'unité centrale alors qu'il y avait un ancien processus CJP dans la machine et le nouveau processus avait une priorité supérieure à celle de l'ancien processus soit (b) il n'y avait pas de processus CJP ayant la commande de l'unité centrale et la tête de la file d'attente des processus prêts a été appelée. En résumé, dans la condition (a) le processus CJP a été retiré du mot de processus en cours RPW et placé dans une liaison de processus PL dans la file d'attente Q/PR/RDY et le nouveau processus NJP, qui se trouvait dans une liaison de processus PL dans la file d'attente Q/PR/RDY a été placé dans le mot RPW intervertissant ainsi les positions des deux processus donnant la commande au nouveau processus NJP qui devient alors le processus CJP et retirant la commande à l'ancien processus. Ensuite, le bloc PCB du processus NJP a été accédé et les informations requises pour le passage en machine du processus NJP, maintenant le processus CJP a été dans la mémoire bloc-notes ou dans le tableau des registres de l'unité centrale CPU.

S'il n'y avait aucun processus CJP ayant la commande de l'unité centrale (condition b) la tête de la file d'attente des processus prêts a été appelée, le processus NJP est devenu le processus CJP du fait que le distributeur a pris le processus NJP de la liaison de processus PL à la tête de la file d'attente des processus prêts et l'a placé dans le mot RPW du fait de l'exécution de cette opération, une liaison de processus PL a été laissée vide dans la file Q/PR/RDY et il est nécessaire de la retirer de la file.

En conséquence, et en commençant maintenant à l'étape de décision 1461 b, la microprogrammerie détermine s'il y avait ou non un processus CJP qui avait la commande de l'unité centrale et dans l'affirmative une liaison de processus libre (FPLS) a été accédée et mise dans la file d'attente et le processus CJP a été décrit. Cependant, s'il n'y avait pas de processus CJP ayant la commande de l'unité centrale, l'octet d'état du mot PMWO du nouveau processus est mis à jour à l'étape 1460 et à nouveau une vérification est effectuée à l'étape 1463 pour déterminer s'il y avait ou non un processus CJP dans la machine. S'il n'y avait pas de processus CJP ayant la commande du processus, la liaison de processus NJP (qui était dans la file d'attente Q/PR/RDY et qui a maintenant la commande de la machine) est retirée de la file Q/PR/RDY à l'étape 1466, et devient un sémaphore de liaison libre FLSP et est alors dans la file d'attente des liaisons de processus libres (805 sur la Fig. 9) et fait alors partie de la file d'attente des liaisons de processus libres à l'étape 1466a, le contenu du registre d'adresse limite

BAR est alors extrait , à l'étape 1464 et le mot de processus en cours RPW du processus NJP (maintenant le processus CJP) situé à l'adresse BAR+56 de la base du système est mis à jour en plaçant l'identification du processus NJP dans le mot RPW à l'étape 1465 . L'indicateur libre est mis à 0, dans le cas où il n'y avait pas de processus CJP. à l'étape 1468. ^{Ensuite} L'indicateur de conflit (c'est-à-dire une bascule de la mémoire auxiliaire 1317a utilisée pour indiquer un conflit éventuel de priorité entre le processus en cours CJP et un processus placé dans la file d'attente des processus prêts) est mis à 0, à l'étape 1467, et l'associateur de segments (mémoire associative AS 132 de la Fig.1) qui est une mémoire adressable est vidée à l'étape 1471, puis le mode de processus est entré à l'étape 1470. (Un mode de processus indique que les exceptions sont traitées par le processus qui est en exécution dans le processeur et non par le système d'exploitation. La microprogrammerie poursuit alors à l'étape CAB 1480 (contrôle du bit d'alarme) et le bit de déroutement asynchrone AB est contrôlé à l'étape 1481, pour déterminer s'il est ou non mis à la valeur 1. Si le bit AB est mis à la valeur binaire 1 un contrôle est effectué , à l'étape 1482 pour déterminer si le numéro d'anneau de processus PRN est supérieur ou égal au numéro d'anneau asynchrone ARN. Les champs AB et ARN sont situés dans l'octet de priorité du bloc PCB de chaque processus et sont significatifs lorsque le processus est dans l'état en cours. Les champs AB et ARN sont obtenus par le mot RPW qui se trouve à l'adresse BAR +56 de la base du système). Les champs AB et ARN à l'adresse BAR+56 du mot RPW sont remis à zéro à l'étape 1483 étant donné que l'étape suivante 1484 consiste dans l'exécution d'une routine de déroutement asynchrone qui doit résoudre les conditions qui ont provoqué la mise en octet d'alarme du bit de déroutement asynchrone ou du numéro d'anneau asynchrone . Si ces bits n'étaient pas remis à zéro, ils donneraient au passage suivant de la microprogrammerie une indication de dérangement alors qu'il n'y aurait rien d'anormal et en conséquence ils provoqueraient toujours la routine de déroutement asynchrone et empêcheraient de passer à l'exécution ^{normale}. Revenant maintenant aux étapes de décision 1481 et 1482 , si le bit AB n'est pas mis à 1 ou si le bit AB est porté au "1" logique et que le numéro PRN n'est pas supérieur au numéro ARN la microprogrammerie entreprend alors de déterminer dans quel mode de processeur fonctionne le mode normal ou le mode émulé. En conséquence le numéro DEXT est contrôlé pour déterminer s'il est ou non mis à zéro et s'il est mis à 0, le mode normal de la machine est utilisé à l'étape 1487 cependant si le numéro DEXT n'est pas mis à 0, le mode émulé est utilisé à l'étape 1486. Le firmware continue en CAB 1480 (contrôle du bit d'alarme) et le bit d'alarme AB est testé pour déterminer si il est ou non mis au 1

binaire en 1481.

Si un programme désire communiquer à un autre programme des renseignements concernant des événements asynchrones, il peut utiliser l'une des 3 voies suivantes : La première consiste à créer différents processus qui accompliront des tâches identiques pour le compte du programme destinataire c'est-à-dire que pratiquement ils attendent sur un sémaphore que les événements asynchrones se produisent et en se référant ensuite au programme destinataire (instruction P). La seconde voie consiste à créer un seul processus qui exécutera sans arrêt un balayage des différents sémaphores afin de recueillir les informations relatives aux événements qui se trouvent sur ces sémaphores. Le processus n'attend plus sur un sémaphore (instruction test P). La troisième voie consiste pour un processus à obliger le processus destinataire à effectuer un déroutement asynchrone à un moment déterminé.

L'avantage de la première voie est évidente : le programme recevra par l'intermédiaire de l'un des processus créés les informations relatives aux événements aussitôt que ceux-ci se produisent. Par contre, cet avantage se paie par la nécessité d'utiliser une zone de mémoire étendue pour les blocs de contrôle des processus créés et pour le logiciel correspondant.

Bien que la voie 2 ne requière qu'une zone de mémoire limitée (1 seul PCB) le processus de collecte des événements doit rester en cours d'exécution jusqu'à ce que le dernier événement se produise. Dans un monoprocesseur où un seul processus est en cours ceci se traduit par une consommation importante de temps machine. La troisième voie a les mêmes avantages que la seconde mais c'est le processus en cours d'exécution qui avertit un ou plusieurs autres processus qu'une information est à prendre en considération. Le problème qui est résolu par l'invention est le suivant : le processus qui est en cours d'exécution CJP, doit prévenir des processus Ji, Pi.....Jn, Pn que quelque chose s'est produit. Or, CJP ne sait absolument pas où sont les autres processus dans quel état ils sont ("suspendu", "prêt" ou "attente"), dans quelle file ils sont placés et quand ils seront susceptibles de reprendre à nouveau la commande du processeur. Par contre il connaît le nom des processus qu'il doit avertir de cet événement.

Par analogie, la situation est la même que celle d'un chercheur dans un laboratoire qui découvre un événement susceptible d'intéresser d'autres chercheurs qui sont actuellement hors du laboratoire. Le premier chercheur connaît le nom des personnes intéressées et sait qu'elles n'entreront dans le laboratoire que lorsque lui-même l'aura quitté mais à un moment quelconque (asynchronisme) après ce départ. Une solution consiste à placer à l'entrée du laboratoire une inscription disant "M. P. veuillez

passer chez le gardien où un message vous attend " lorsque le chercheur J.P passera devant la loge du gardien il sera informé qu'un message l'attend et pourra prendre connaissance de ce message.

- Le mécanisme du bit d'alarme est identique en ce que chaque fois qu'un nouveau processus J.P prend la commande du processeur, le bit d'alarme ou bit de déroutement asynchrone est testé. Si il est à 1, le numéro d'anneau est également testé. Si il est supérieur ou égal à un numéro d'anneau d'alarme (c'est-à-dire si le niveau de privilège du processus est inférieur à celui de l'anneau d'alarme de déroutement asynchrone est effectué). Ce dernier test correspond au désir de ne pas dérouter un processus qui est appelé à traiter des fonctions système (privilège 0). En fait, il est assez peu probable qu'un événement signalé par un processus puisse concerner le fonctionnement du système lui-même et la précaution qui est ainsi prise est en général superflue.
- 15 Le déroutement asynchrone se traduit par une interruption ou une suspension du processus et par une entrée dans le manipulateur d'exceptions et a accès à la table d'exceptions.

- Le micrologiciel a écrit le message à communiquer au processus soit en mémoire soit dans des registres généraux du système. L'accès du processus au message se fait selon la procédure de traitement des exceptions qui a été décrite dans la demande de brevet française N° 73 42700 du 30.11.73 intitulée : " Dispositif et méthode assurant le traitement des exceptions dans une machine de traitement de données", déposée par la demanderesse, c'est-à-dire que l'exception de déroutement asynchrone exécute par l'intermédiaire d'une instruction V simulée une préparation de pile et d'entrée ou un mécanisme d'interruption qui procède à un échange de processus.
- 20 25

- Le bit d'alarme est testé chaque fois que un nouveau processus prend la commande du processeur. Il doit aussi être testé au cours des instructions de sortie EXIT lorsque le niveau de privilège a été modifiée lors de l'exécution du processus. Le niveau de privilège du processus peut être devenu supérieur au niveau de l'alarme et le message intéresse alors le processus.
- 30

- Il est bien entendu possible d'appliquer ce procédé à la gestion d'événements et d'en faire un emploi généralisé dans un système à processeurs multiples sans sortir pour cela du cadre de l'invention.
- 35

REVENDECATIONS

1. Procédé de communication d'informations entre plusieurs processus pour un système informatique comprenant un processeur central relié à une mémoire centrale, un bloc de contrôle de processus étant mis en mémoire pour chacun des processus, le processeur central étant à chaque instant commandé par un processus alors que les autres processus sont dans des états dits "prêt", "suspendu" ou "attente" organisés en files, un distributeur donnant la commande du processeur au processus "prêt" de priorité la plus élevée, un niveau de privilège symbolisé par un numéro d'anneau étant d'autre part affecté à un moment donné à chaque processus, caractérisé en ce que chaque bloc de contrôle de processus contient au moins un élément de mémorisation spécialisé, des moyens permettant au processus en cours d'exécution de modifier la position dudit élément, le système comprenant des moyens de contrôle dudit élément, une position binaire définie de celui-ci provoquant un déroutement pris en charge par un manipulateur d'exceptions qui permet de communiquer un message au processus.

2. Procédé selon la revendication 1 caractérisé en ce que le contrôle de l'élément de mémorisation précité est effectué chaque fois que le distributeur attribue la commande du processeur à un nouveau processus.

3. Procédé selon la revendication 1, caractérisé en ce que un système de privilège étant établi en fonction de la nature des opérations effectuées, le niveau de privilège étant défini par un numéro d'anneau, le test de l'élément de mémorisation précité est suivi du test du numéro d'anneau dans lequel le processus travaille, le déroutement précité n'étant autorisé que dans le cas où ledit numéro PRN est supérieur ou égal à un numéro d'anneau d'alarme ARN.

4. Procédé selon la revendication 3, caractérisé en ce que un test de l'élément de mémorisation précité est effectué lors de l'exécution d'une instruction de sortie du processeur, lorsque le processus a changé au cours de son exécution.

5. Dispositif de mise en oeuvre du procédé selon la revendication 1, caractérisé en ce qu'il comprend des moyens de test de l'élément de mémorisation, des moyens de test du numéro d'anneau, et des moyens d'appel d'une procédure de déroutement asynchrone.

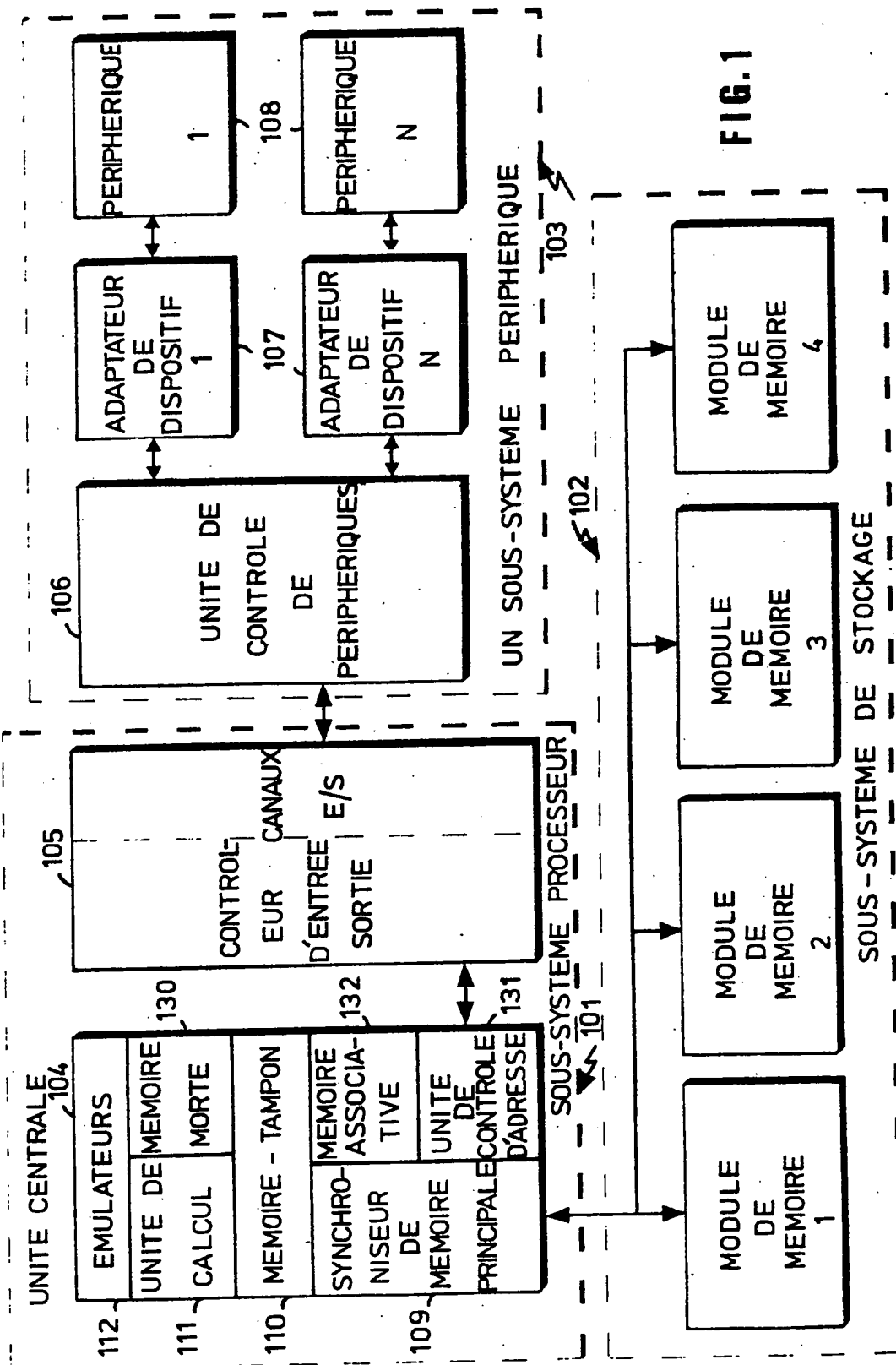


FIG.1

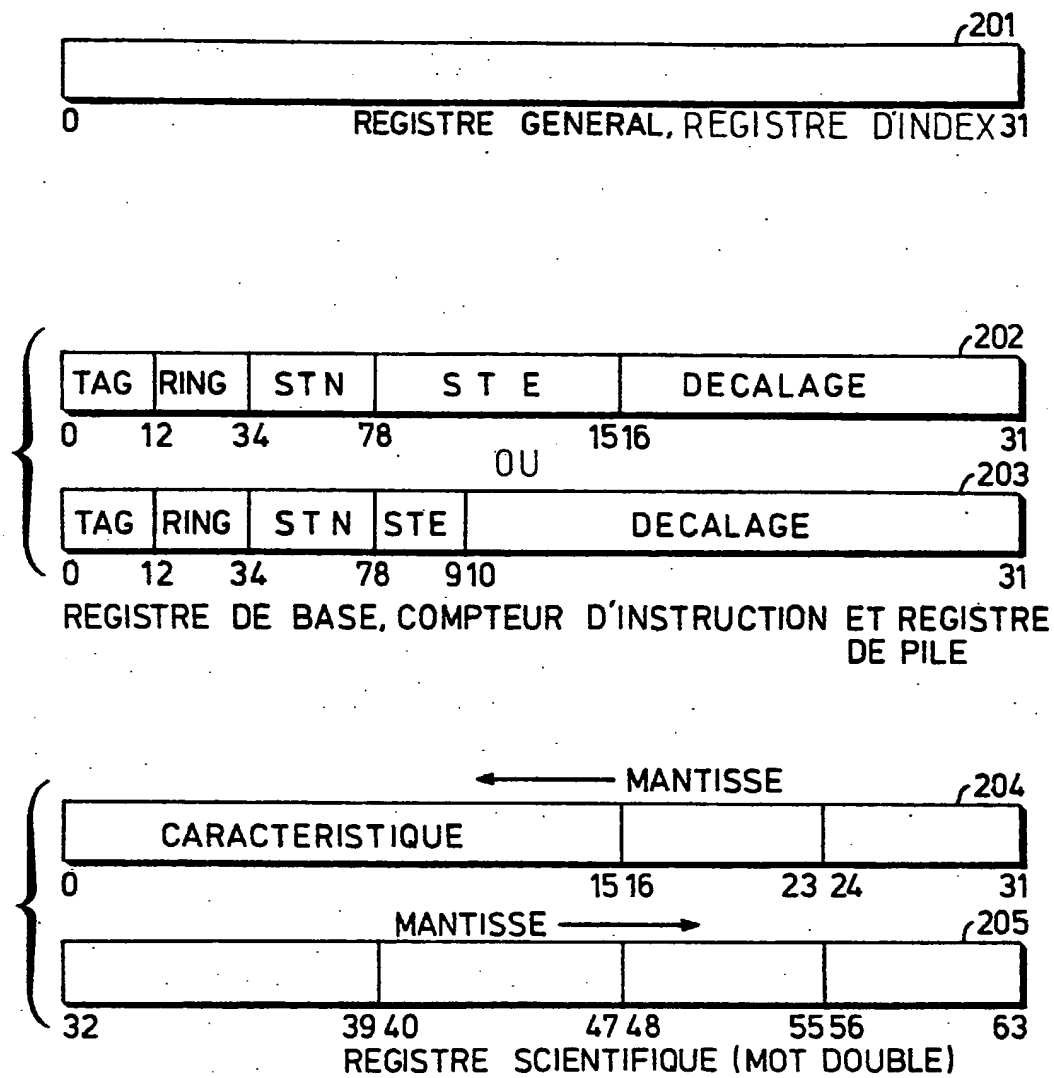


FIG. 2

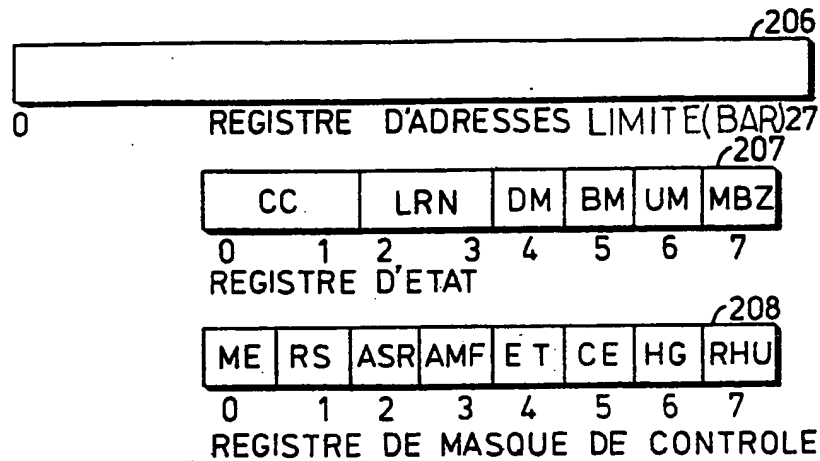


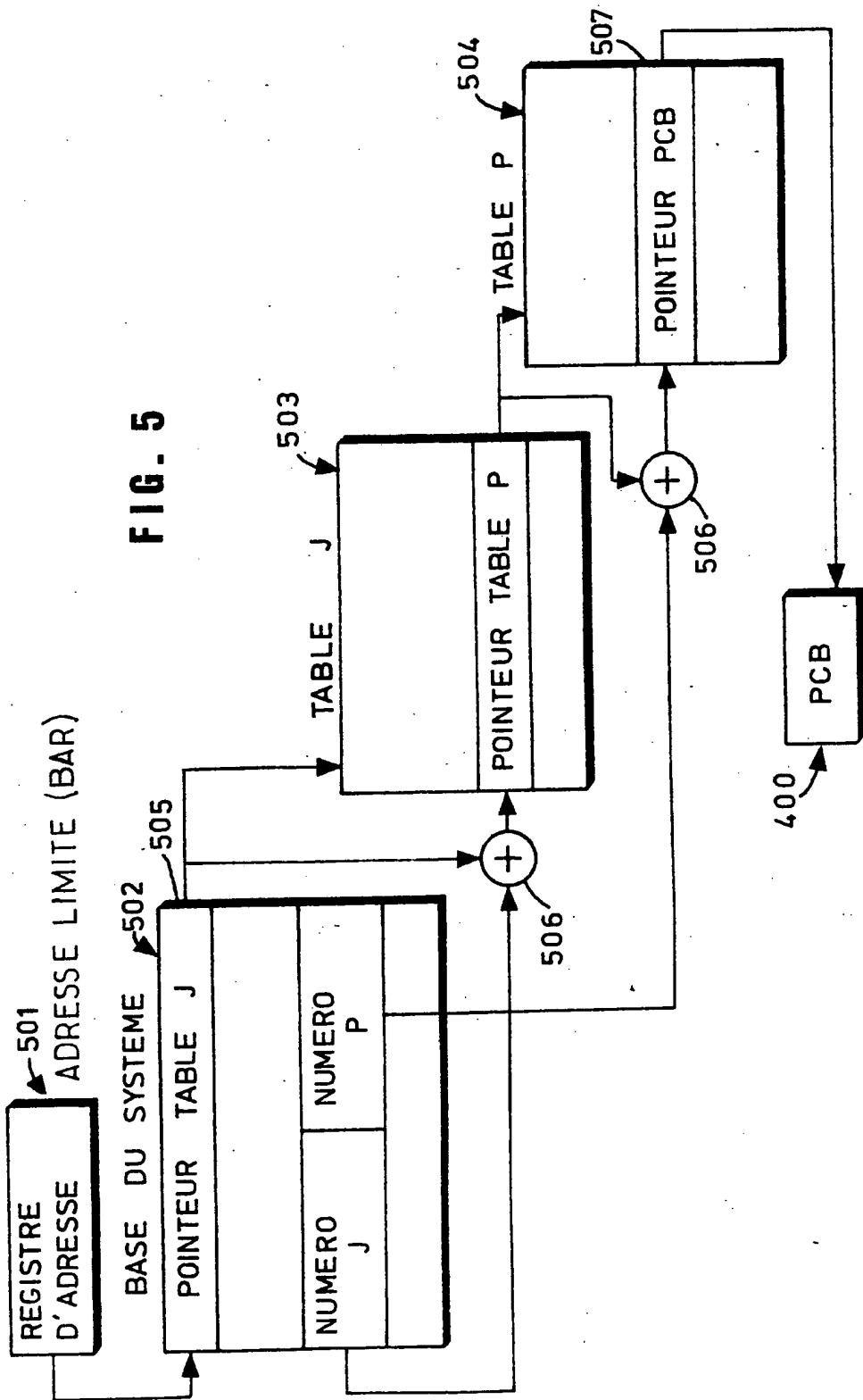
FIG. 3

LEGENDE DES ZONES DE STOCKAGE RESERVEES
DANS LES REGISTRES.

AMF = MECANISME COMPTAGE
ASR = RECONFIGURATION AUTOM.
MEMOIRE.
BM = MASQUE BINAIRE
CC = CODE CONDITION
CE = ERREUR CORRIGEE
DM = MASQUE DECIMAL
ET = TOLERANCE ENVIRONNEMENT
HG = PORTE DU MATERIEL
LRN = DERNIER NUMERO ANNEAU
MBZ = DOIT ETRE A ZERO
ME = ERREUR MACHINE
RING = NUMERO D'ANNEAU
RHU = RESERVE AU MATERIEL
RS = RELANCE REUSSIE
S = SIGNE
STE = ELEMENT TABLE SEGMENTS
STN = NUMERO TABLE SEGMENTS
TAG = TYPE DESCRIPTEUR
UM = MASQUE DEPASSEMENT NEGATIF

| | | | | | | | | |
|------------------------------|---|---------------------------|----------|------|------|-------------|-----|-------------|
| EMPLACEMENT DE MEMOIRE | -40 | HEURE DEBUT COURANTE | | | | NOM | CET | } EN OPTION |
| | -32 | COMPTAGE TEMPS PRET | | | | RTA | | |
| | -24 | COMPTAGE TEMPS D'ATTENTE | | | | WTA | | |
| | -16 | COMPTAGE EXECUTION TOTALE | | | | RUA | | |
| ADRESSE PCB | -8 | TEMPS D'EXECUTION PARTIEL | | | | RTO | | |
| | 0 | FONCTION | PRIORITE | ETAT | DEXT | | | |
| | 4 | ETAT | MBZ | MP | MBZ | | | |
| | 8 | | | | | | | |
| | 12 | DETSZ | DETA | | | | | |
| | 16 | STWSZ | STWA | | | ASW0 | | |
| | 20 | STWSZ | STWA | | | ASW1 | | |
| | 24 | | | | | | | |
| | 28 | CONTENU DE T | | | | | | |
| | 32 | CONTENU DE IC | | | | | | |
| | 36 | MBZ | | | | | | |
| | 40 | | | | | | | |
| | 44 | | | | | | | |
| | 48 | | | | | | | |
| 52 | ZONE DE MEMORISATION DES REGISTRES DE BASE (8 MOTS) | | | | | | | |
| 80 | | | | | | | | |
| 84 | ZONE DE MEMORISATION DES REGISTRES GENERAUX (16 MOTS) | | | | | | | |
| 144 | | | | | | | | |
| 148 | ZONE DE MEMORISATION DES REGISTRES SCIENTIFIQUES (8 MOTS) | | | | | } EN OPTION | | |
| 176 | | | | | | | | |

FIG. 4



ADRESSE ABSOLUE 0

BAR

600

BAR+44

BAR+48
+52BAR+56
+60+64
+68

+72

| | | |
|--|---------------------------------------|--------------------------------|
| RESERVE AU MATERIEL | | |
| MOT TABLE J | | |
| MOT TABLE G | | |
| CELLULE EXCEPTION SYSTEME N°0 | | |
| .. | .. | N°1 |
| .. | .. | N°2 |
| .. | .. | N°3 |
| .. | .. | N°4 |
| .. | .. | N°5 |
| .. | .. | N°6 |
| .. | .. | N°7 |
| .. | .. | N°8 |
| CELLULE EXCEPTION CANAL | | |
| MOT FILE INTERNE PROCESSEUR | | |
| NFS | COMPTAGE RELANCE INITIAL | COMPTAGE RELANCE COURANT |
| MOT PROCESSUS EN COURS | | |
| ATSZ | POINTEUR DE TABLE D'ABSOLUTISATION | |
| NUMERO SERIE PROCESSEUR | | |
| LIMITE SUPERIEURE MEMOIRE PRINCIPALE | | |
| DISPO.CHARGEMENT CN N° | | DISPOSITIF CN N° |
| RSU | TYPE DISPOSITIF MATERIEL | SOUS.TYPE |
| RSU | TYPE DISPO. CHARGEMENT INITIAL | SOUS.TYPE |
| CELLULE RELANCE | | |
| RESERVE A L'EXTENTION MULTIPROCESSEUR | | |

BASE DU SYSTEME

FIG.6

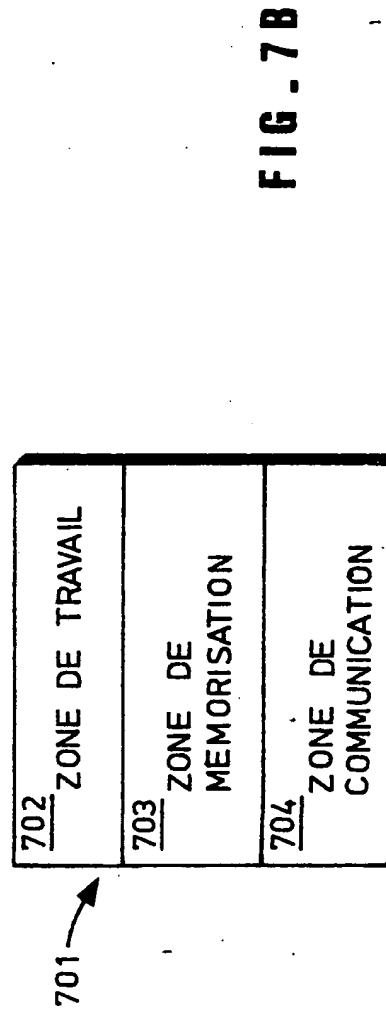
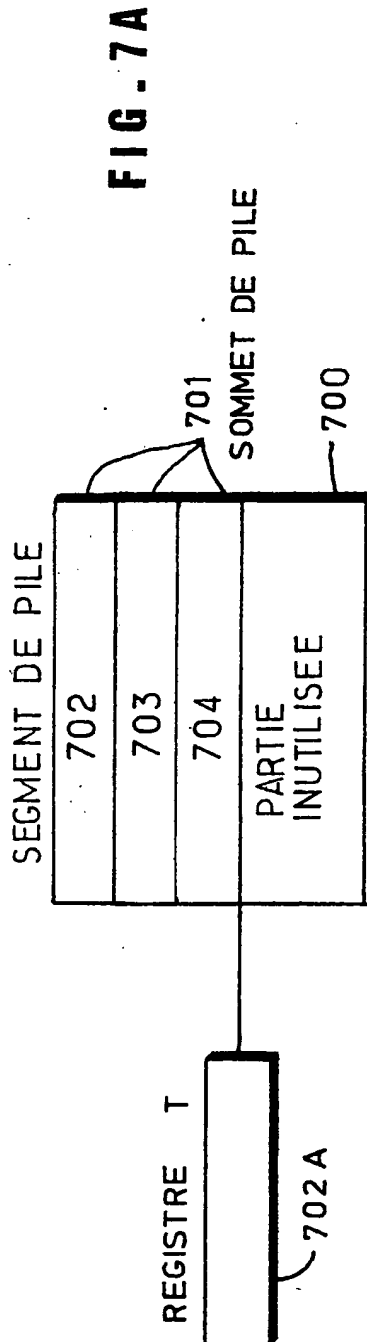
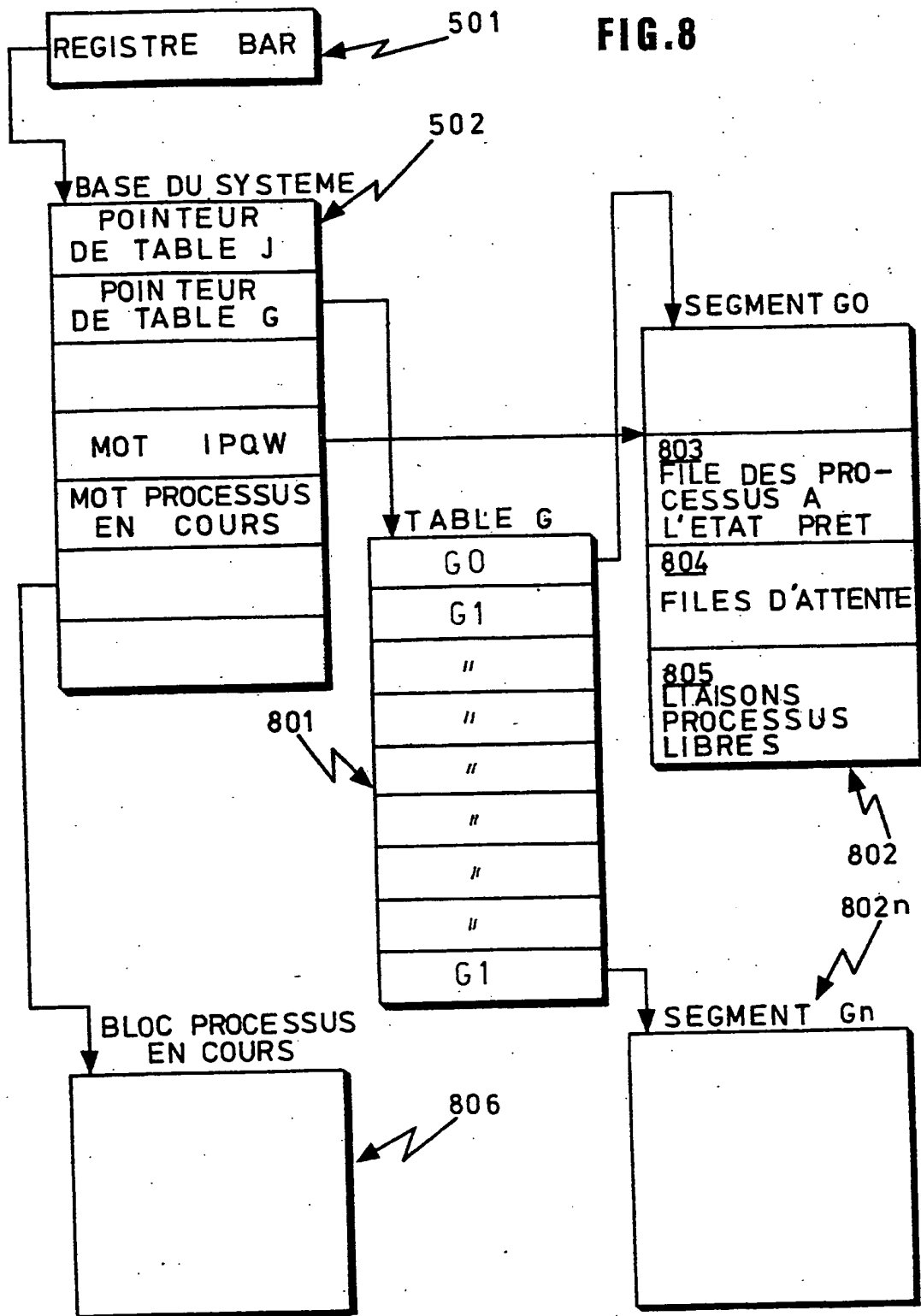
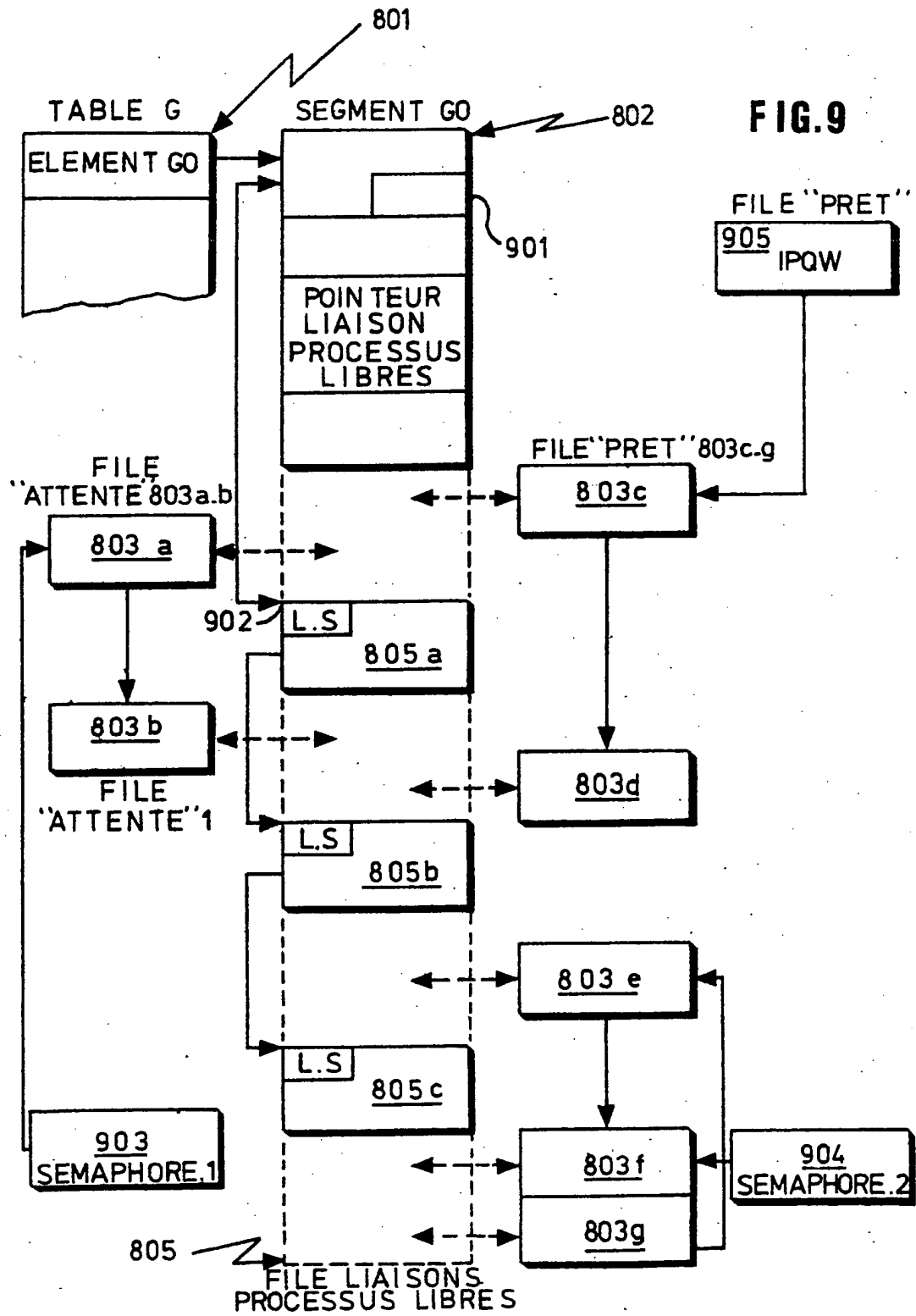


FIG.8





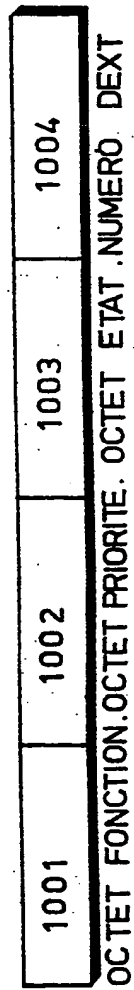


FIG.10a

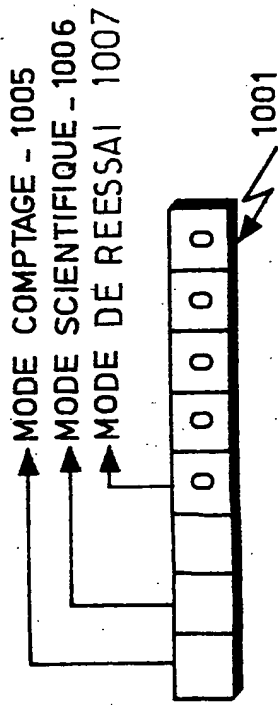


FIG.10b

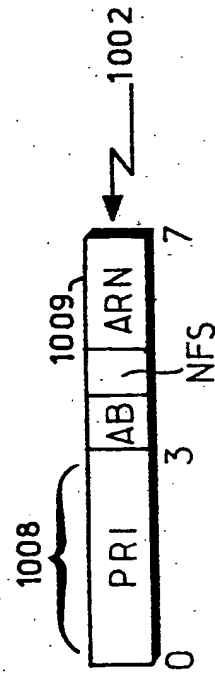
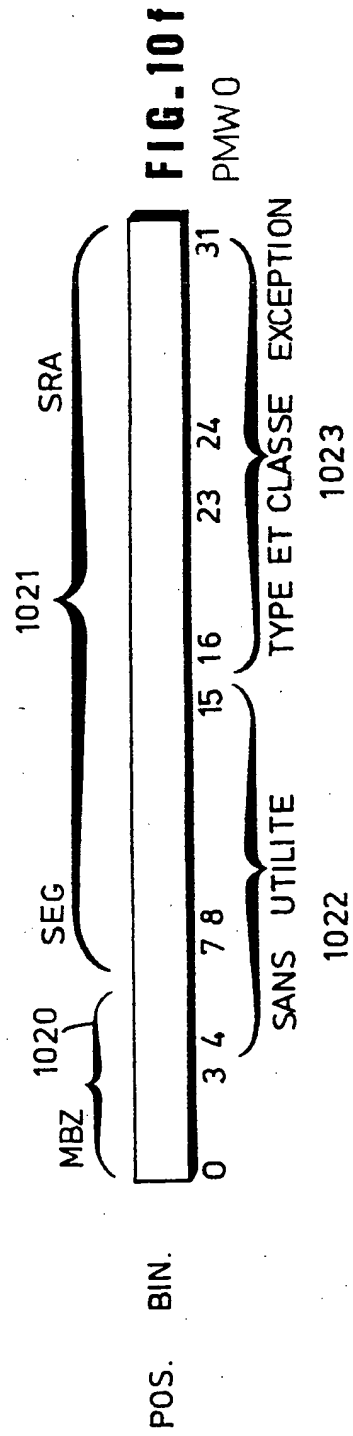
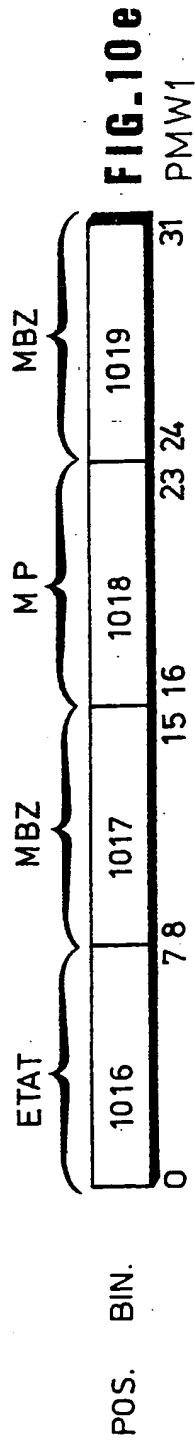
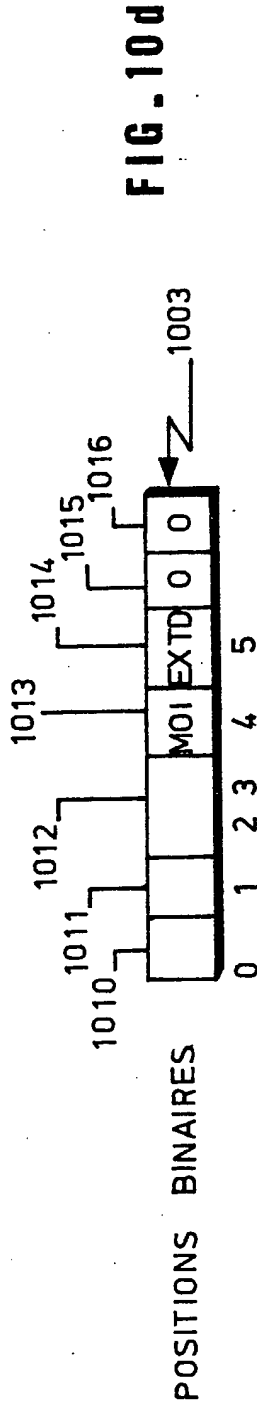
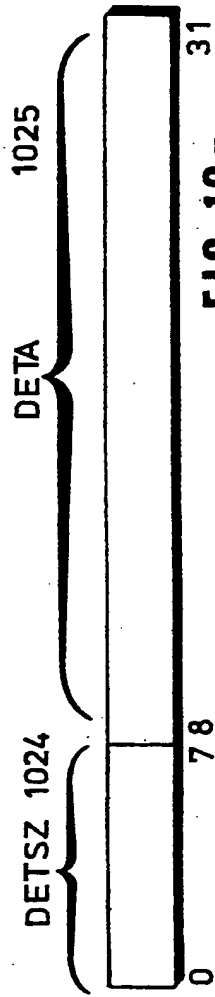


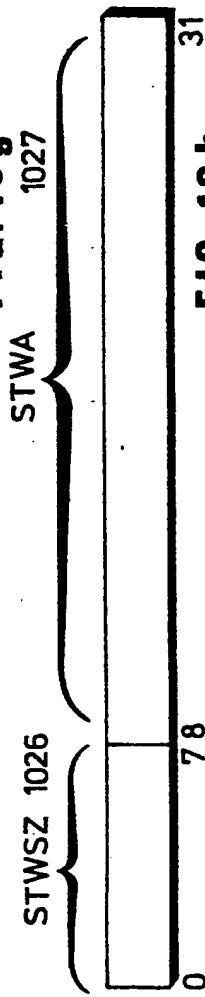
FIG.10c



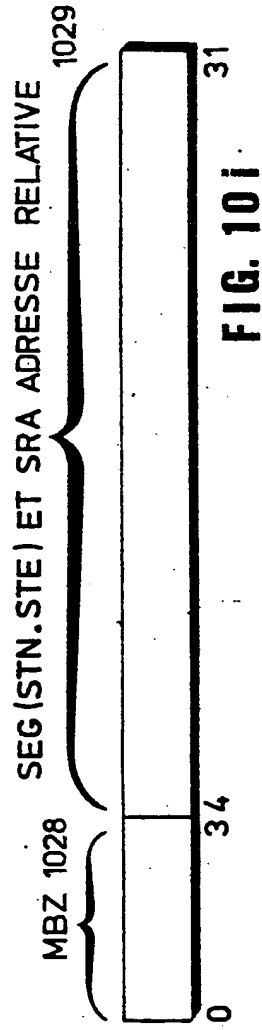
POSITIONS BINAIRES

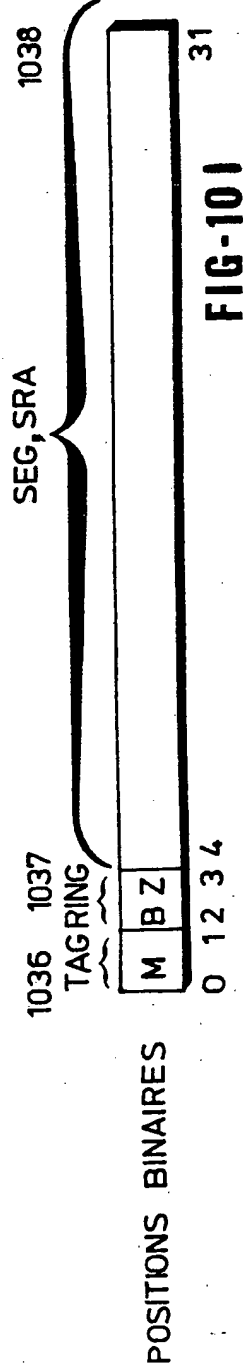
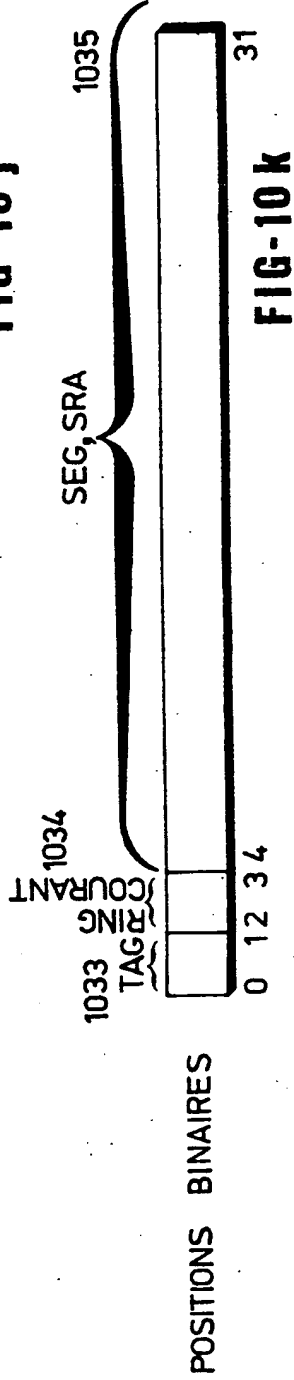
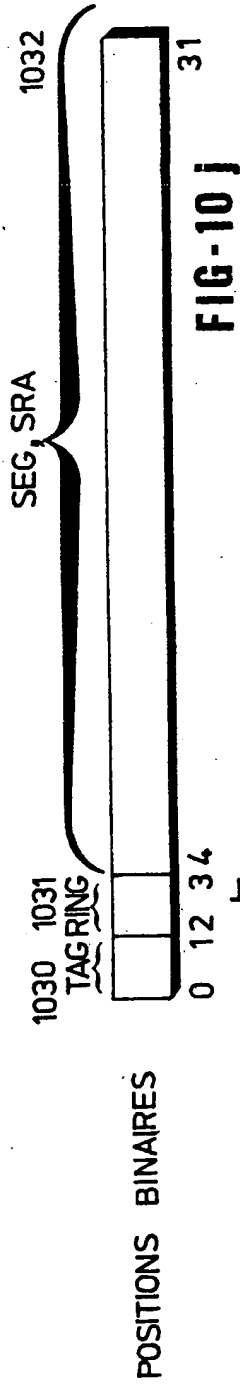


POSITIONS BINAIRES



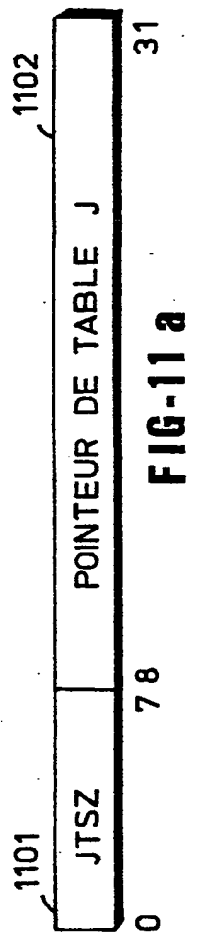
POSITIONS BINAIRES



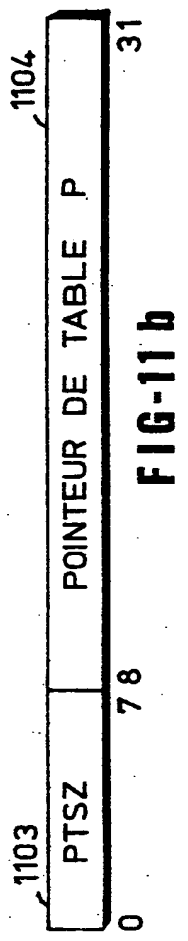


PL : ~~XIV~~ : 35

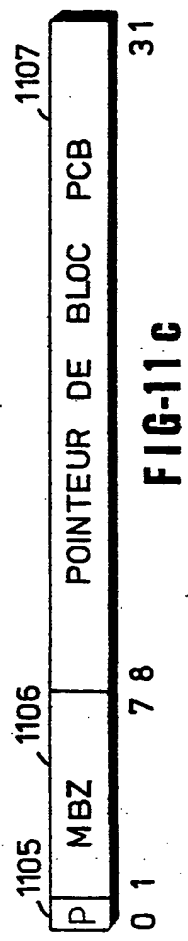
2269150



POSITIONS BINAIRES



POSITIONS BINAIRES



POSITIONS BINAIRES

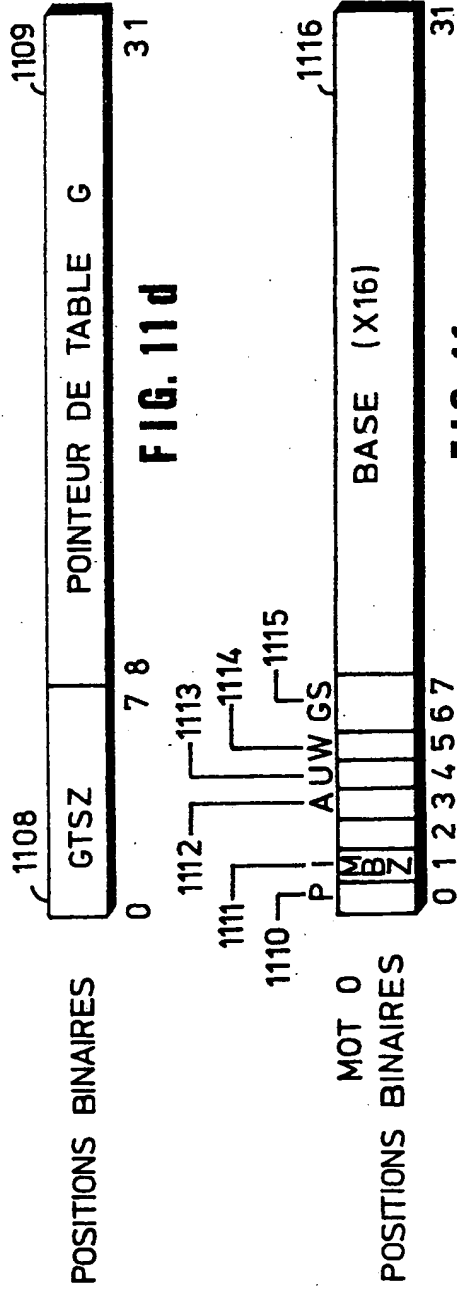
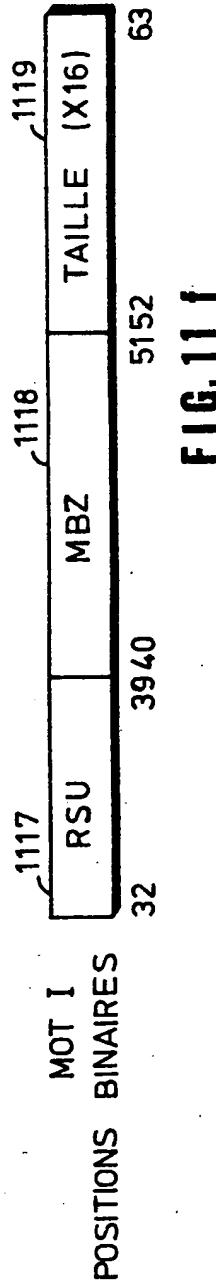
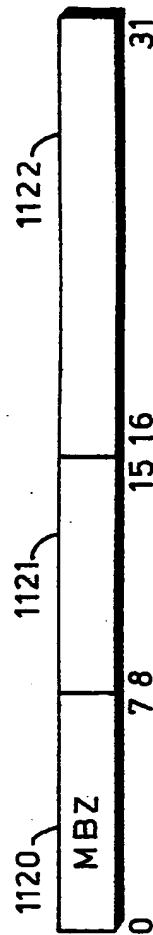


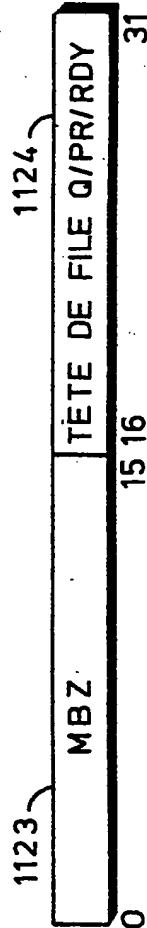
FIG. 11e





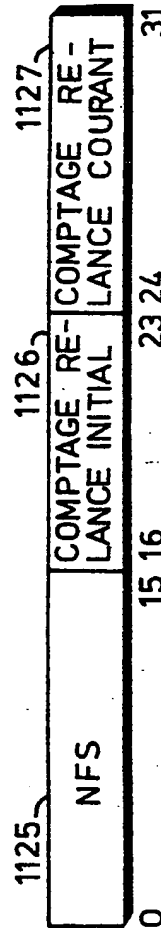
POSITIONS BINAIRES

FIG. 11g



POSITIONS BINAIRES

FIG. 11h



POSITIONS BINAIRES

FIG. 11i

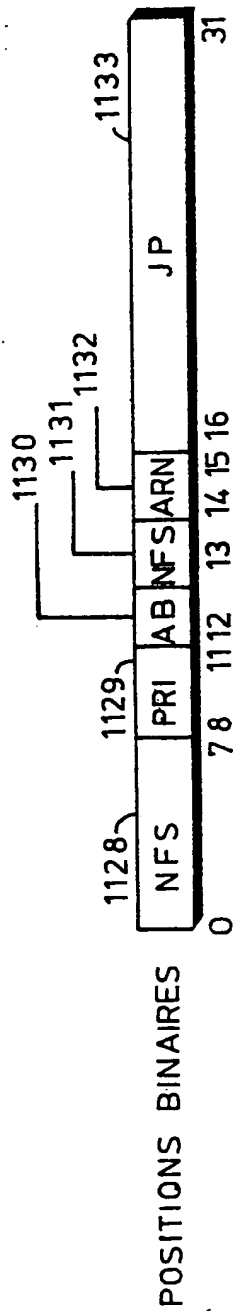


FIG. 11j

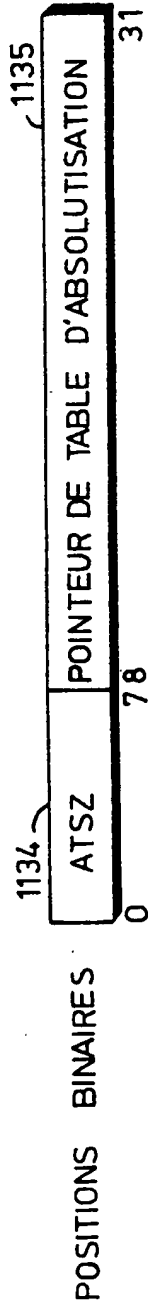
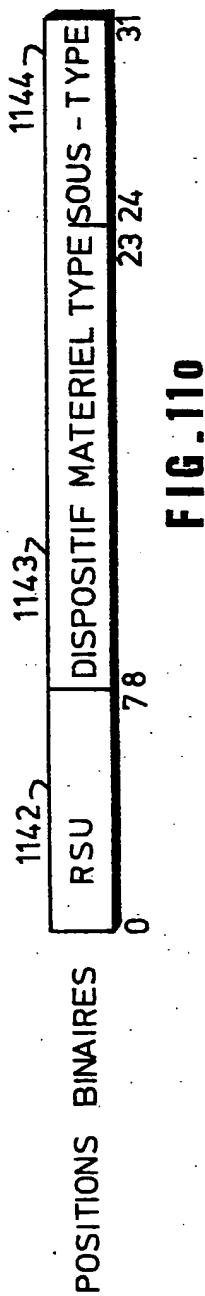
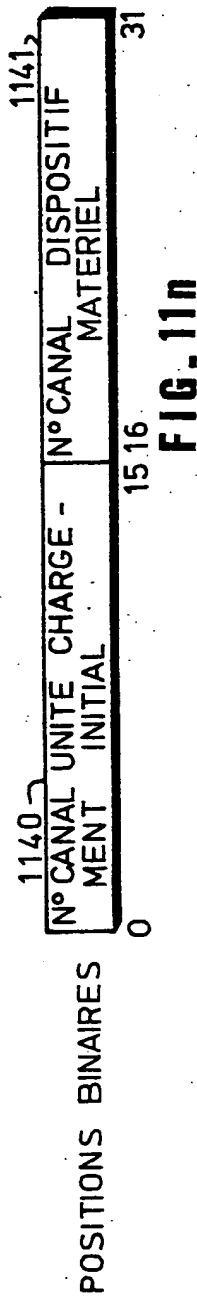
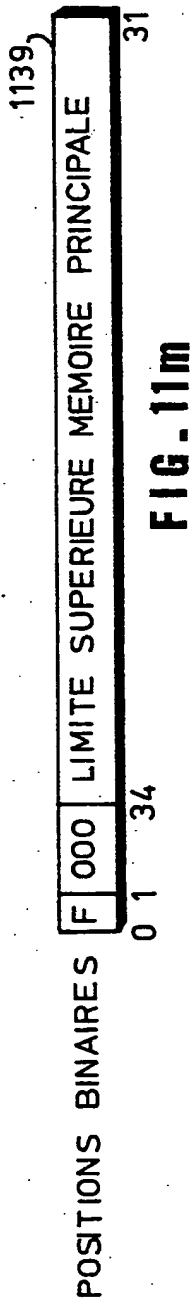


FIG. 11k



FIG. 11.l



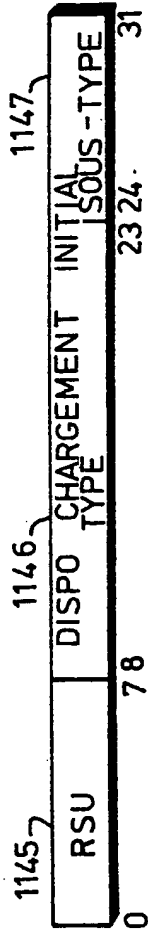


FIG. 11p

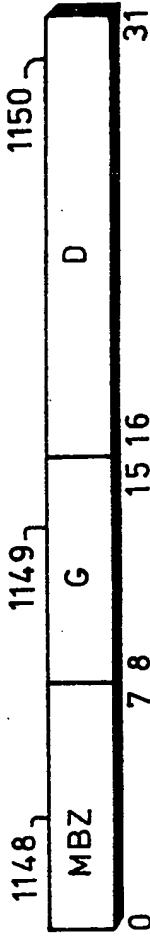


FIG. 11q

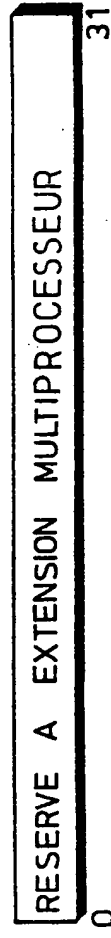
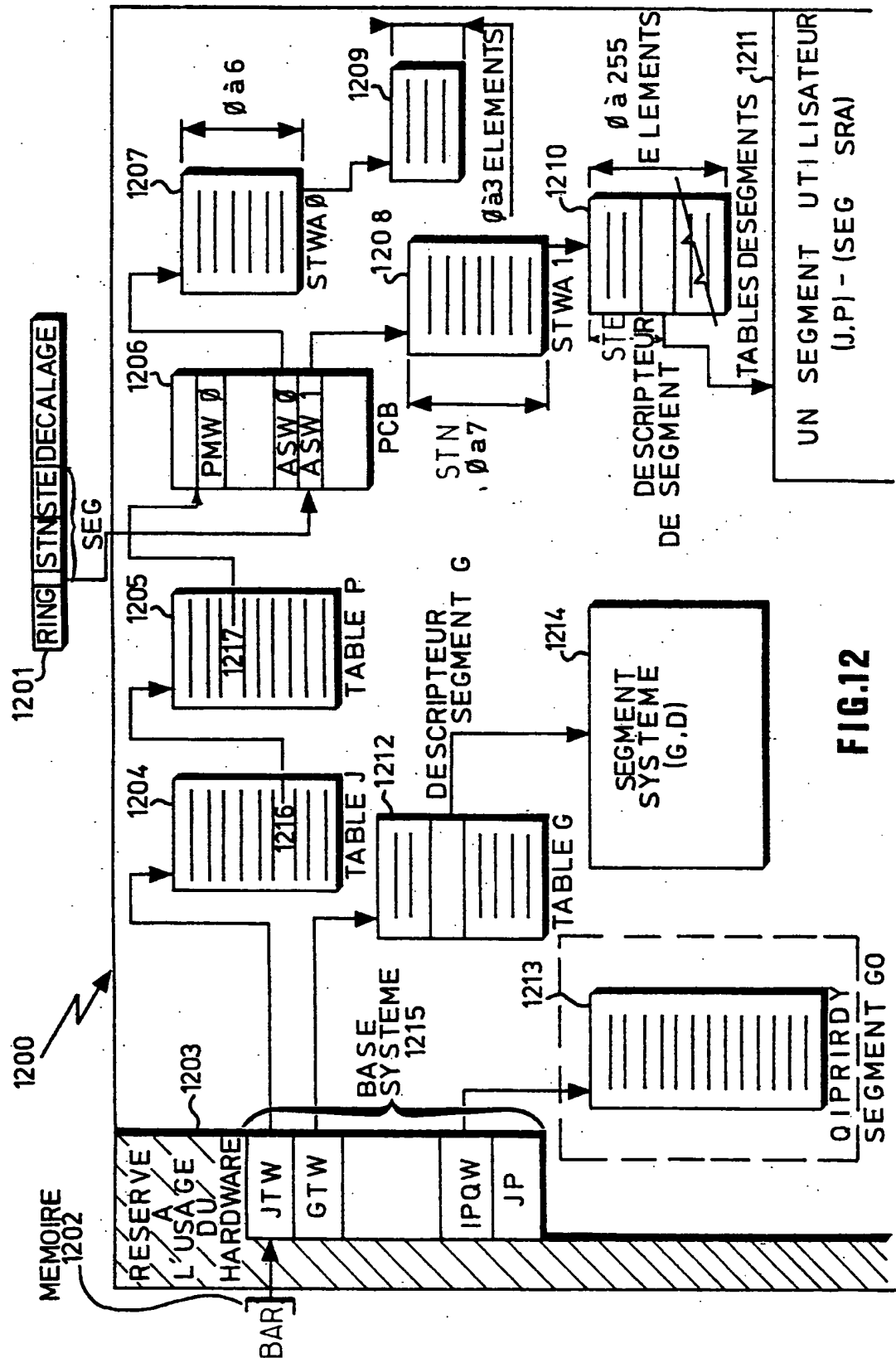


FIG. 11r

POSITIONS BINAIRES

POSITIONS BINAIRES

POSITIONS BINAIRES



PROCESSEUR CENTRAL (CPU) 1306

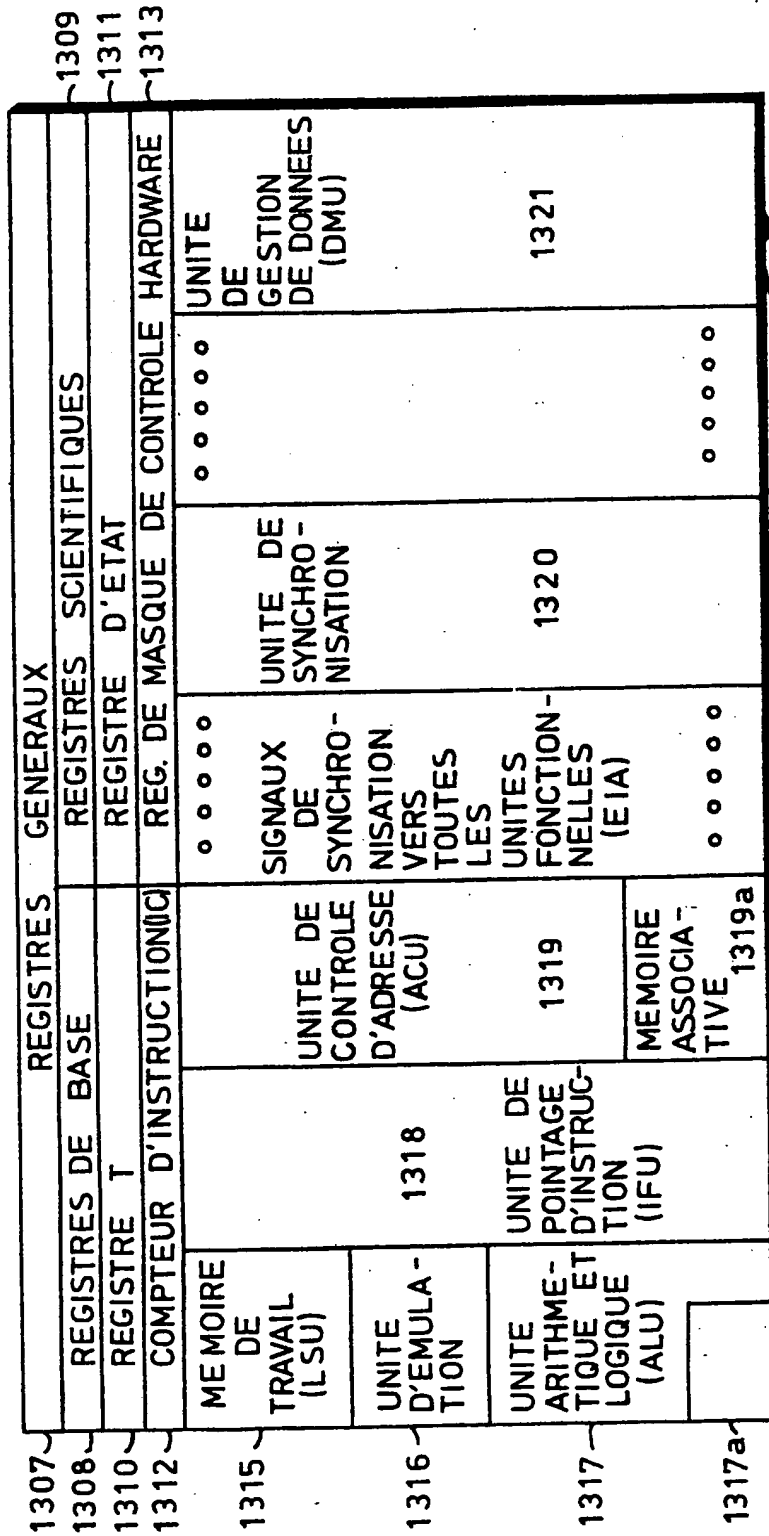


FIG. 13a1

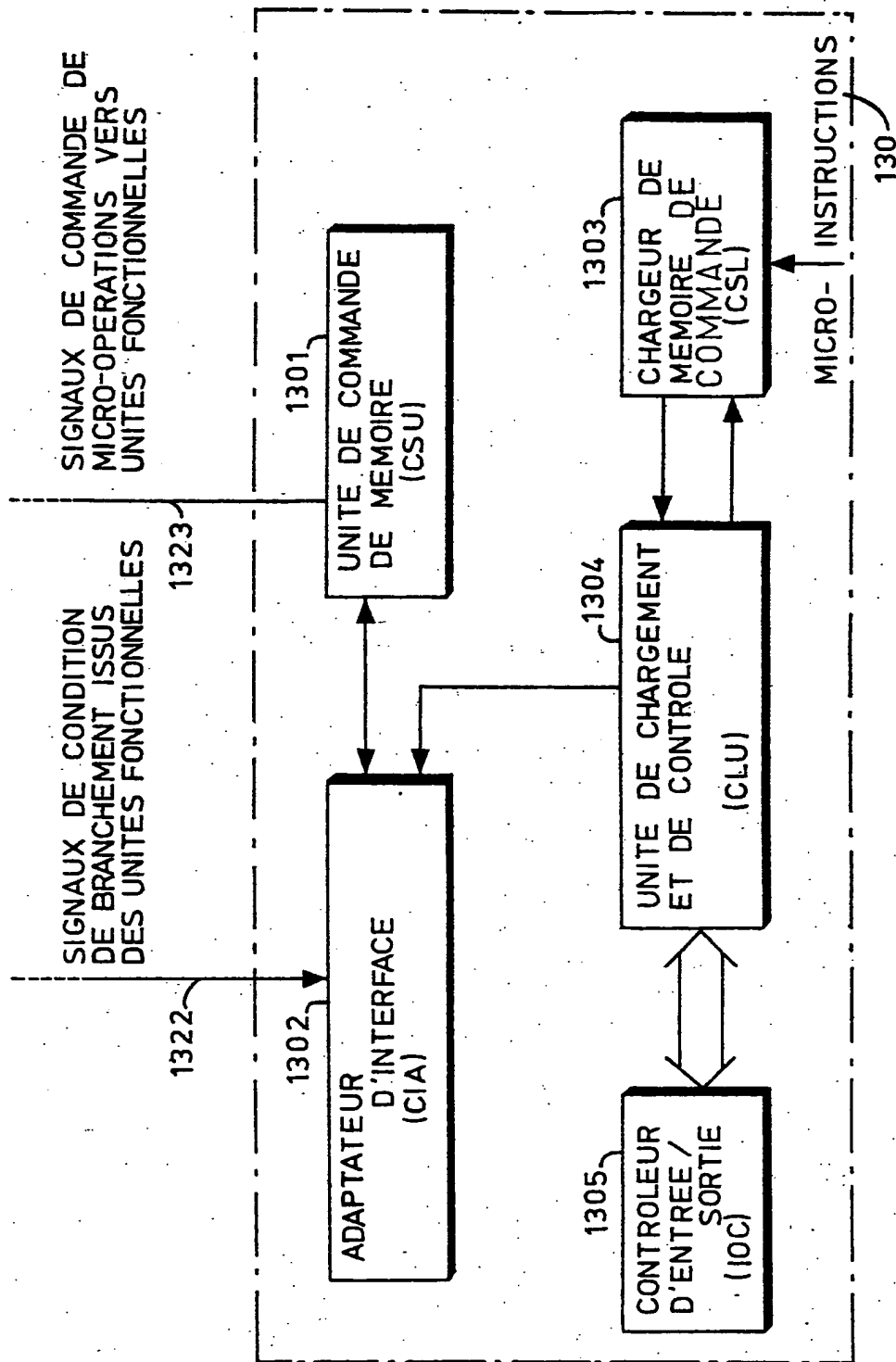
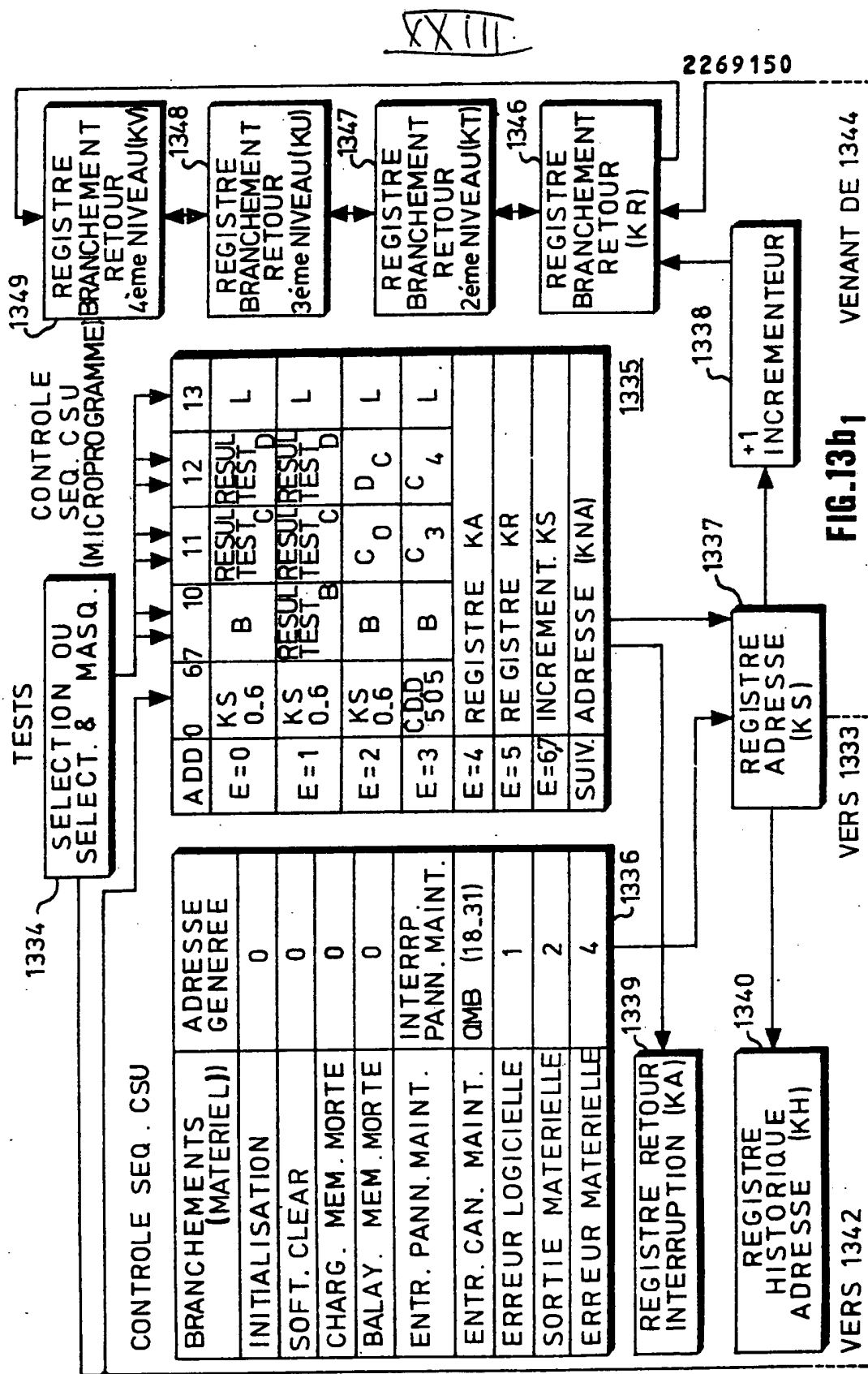


FIG. 13a₂



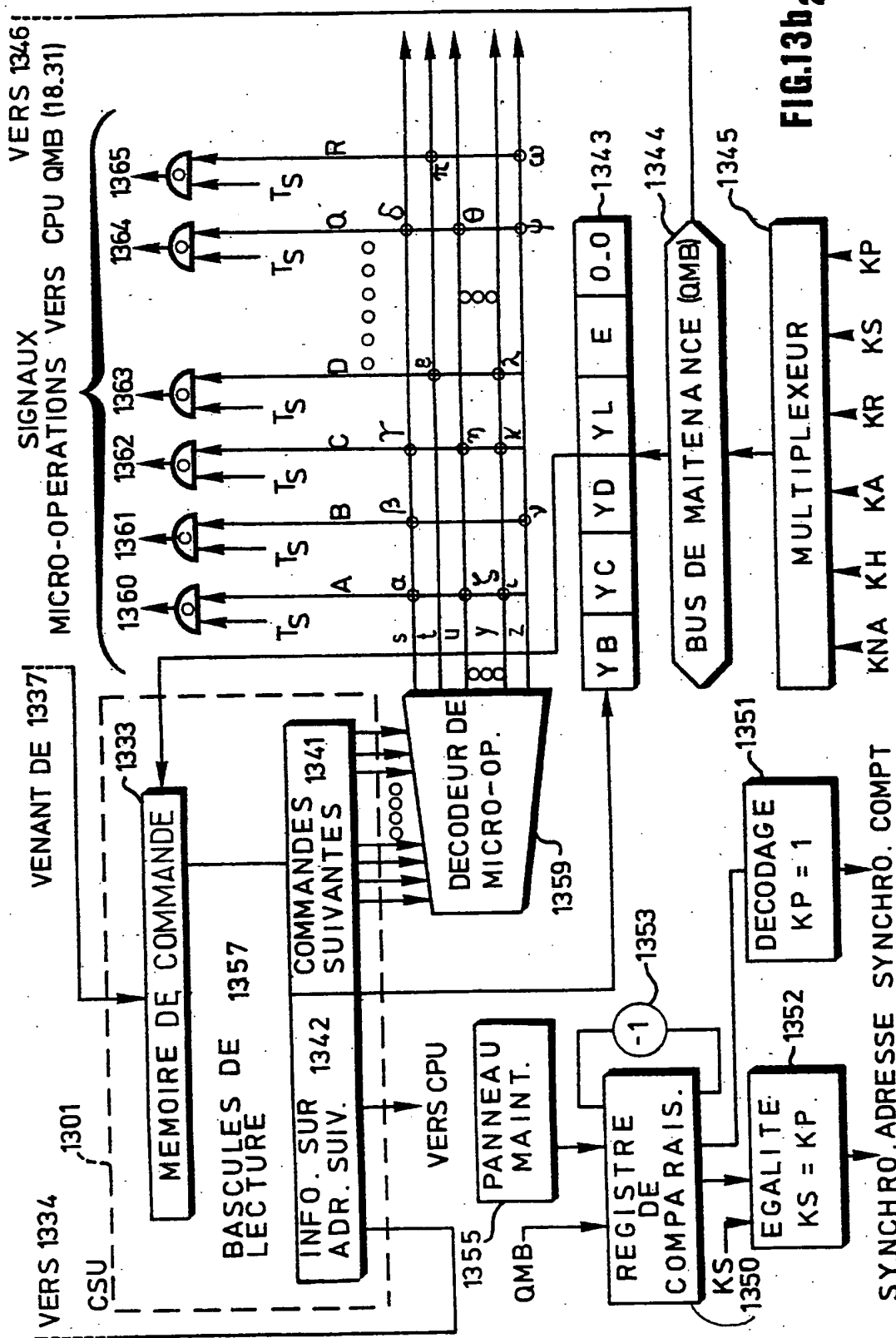


FIG.13b₂

PL: ~~XXV~~: 35

2269150

| UTILISATION ZONES | 1326 TYPE DE SEQUENCE | 1327 BRANCHEMENT ET/OU MICRO OPERATIONS | 1328 GENERATION ET DESIGNATION DES CONSTANTES | 1329 DONNEES VERS BUS | 1330 MICRO - OPERATIONS | 1331 CONTRÔLE |
|----------------------|-----------------------------|---|--|--------------------------------|-------------------------------|------------------|
| NOM DE ZONE | E | A B C D L | K | QA QB | F | P |
| BITS | 3 | 6 4 6 6 1 | 14 | 4 4 | 32 | 4 |
| | | 23 | | 8 | | |

FIG. 13C

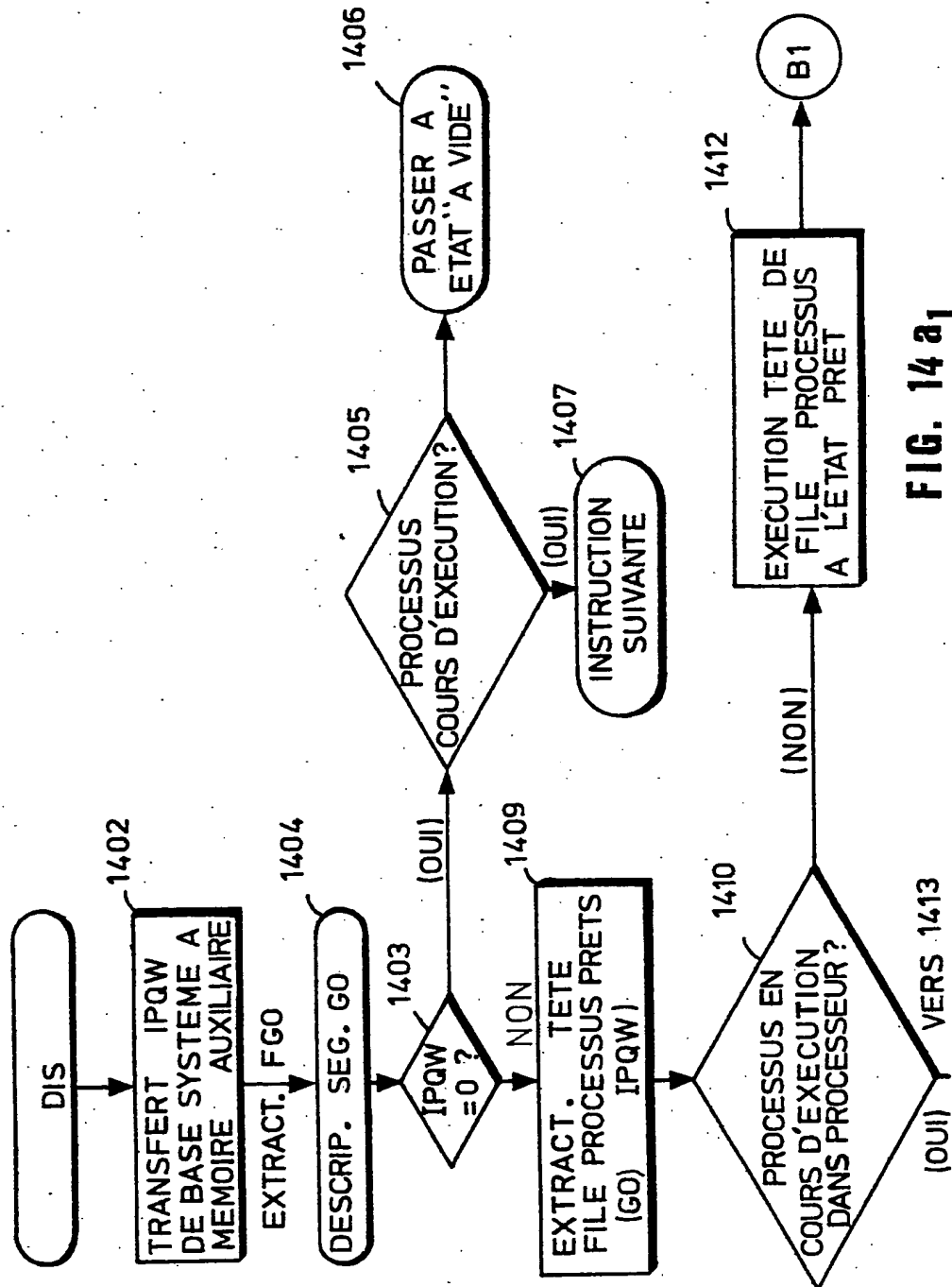


FIG. 14a1

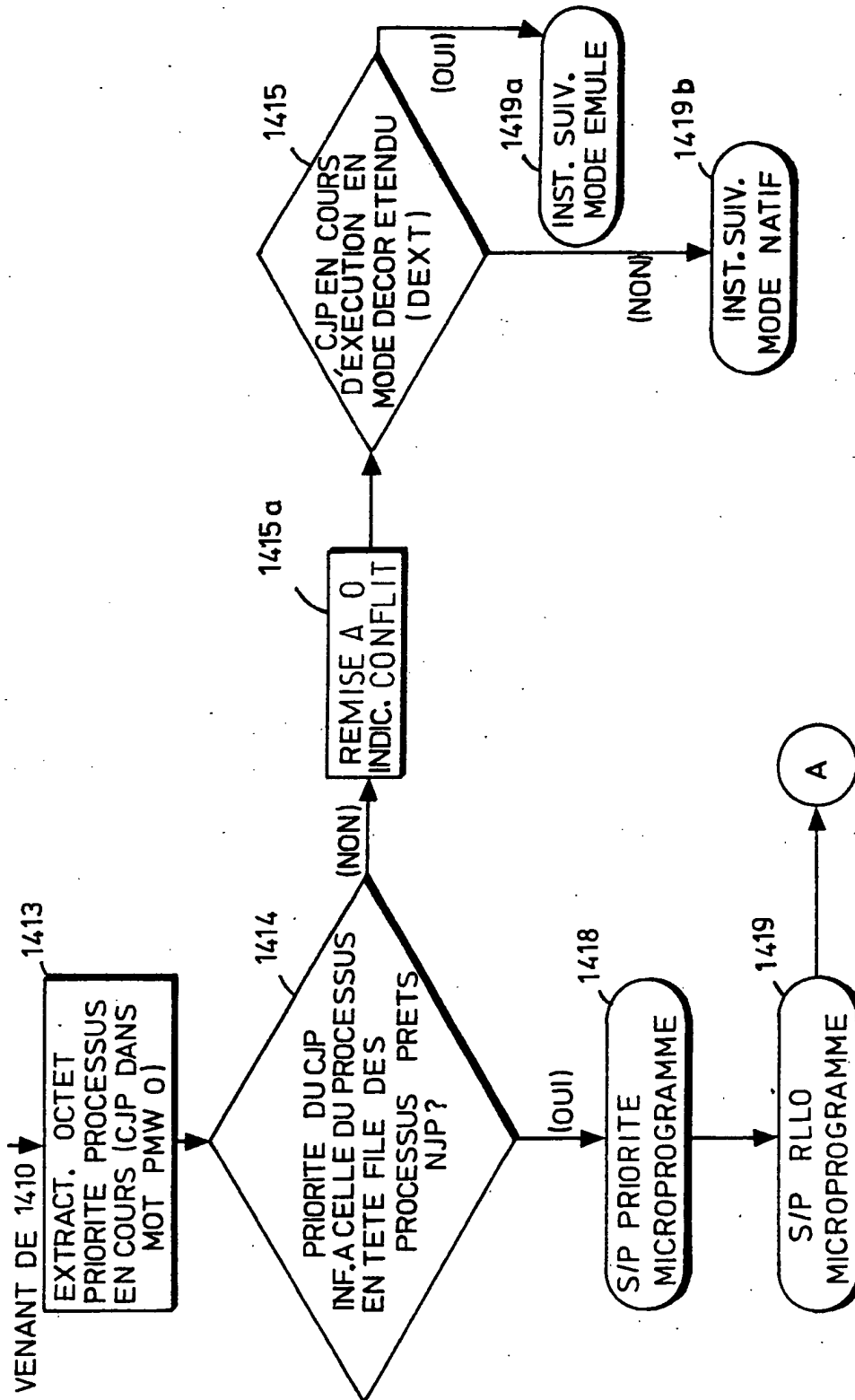


FIG. 14a2

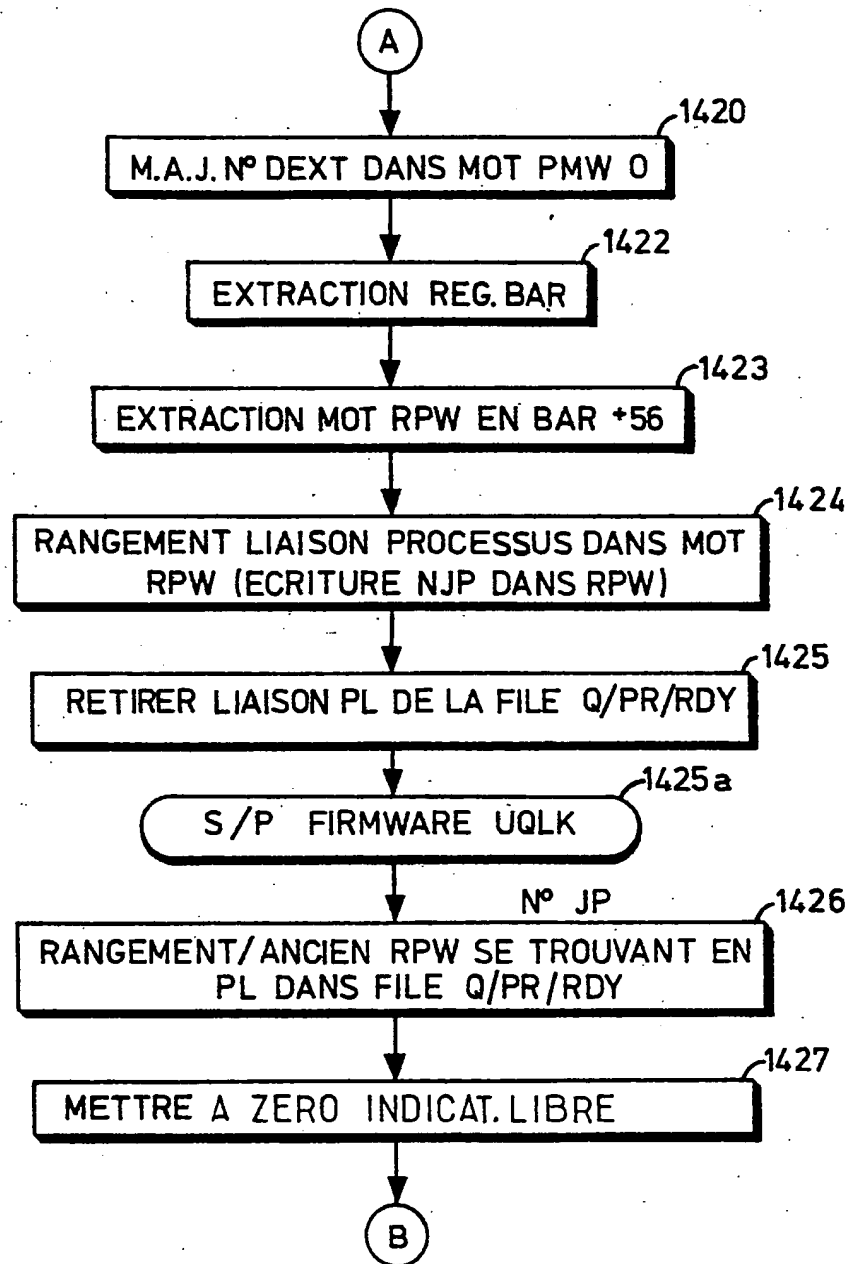


FIG. 14b

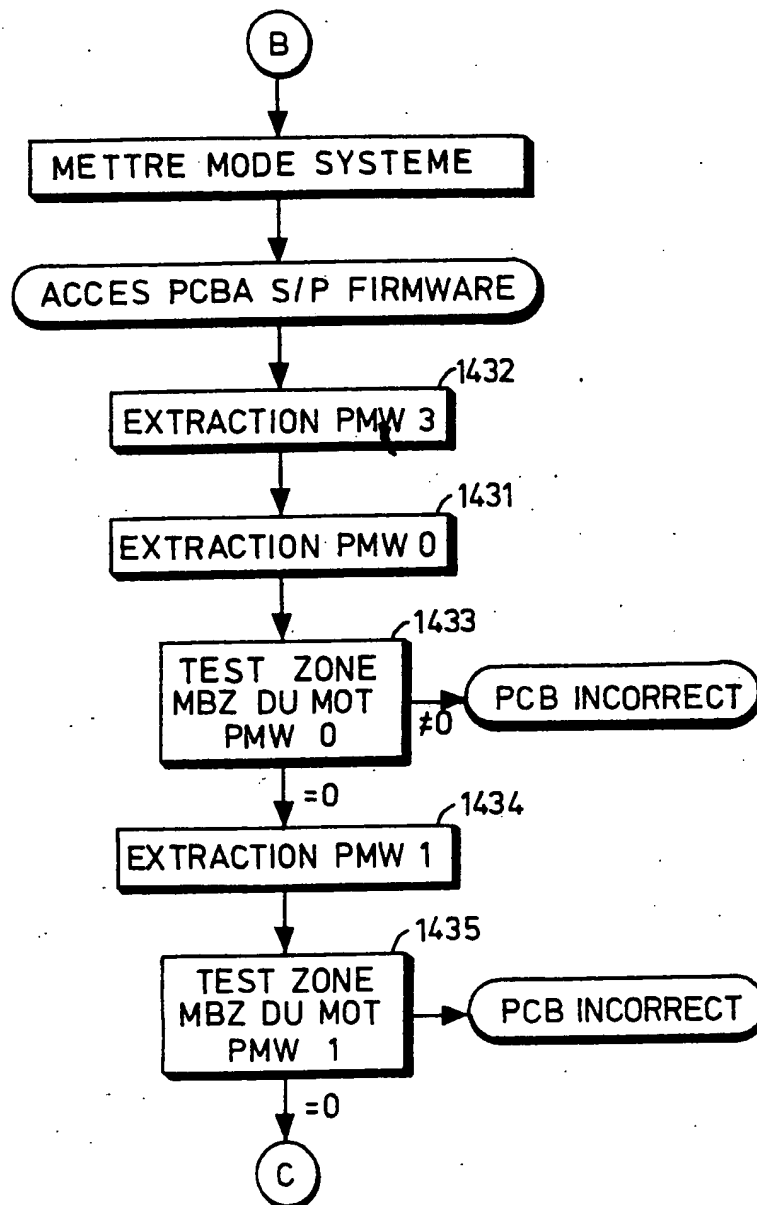


FIG. 14c

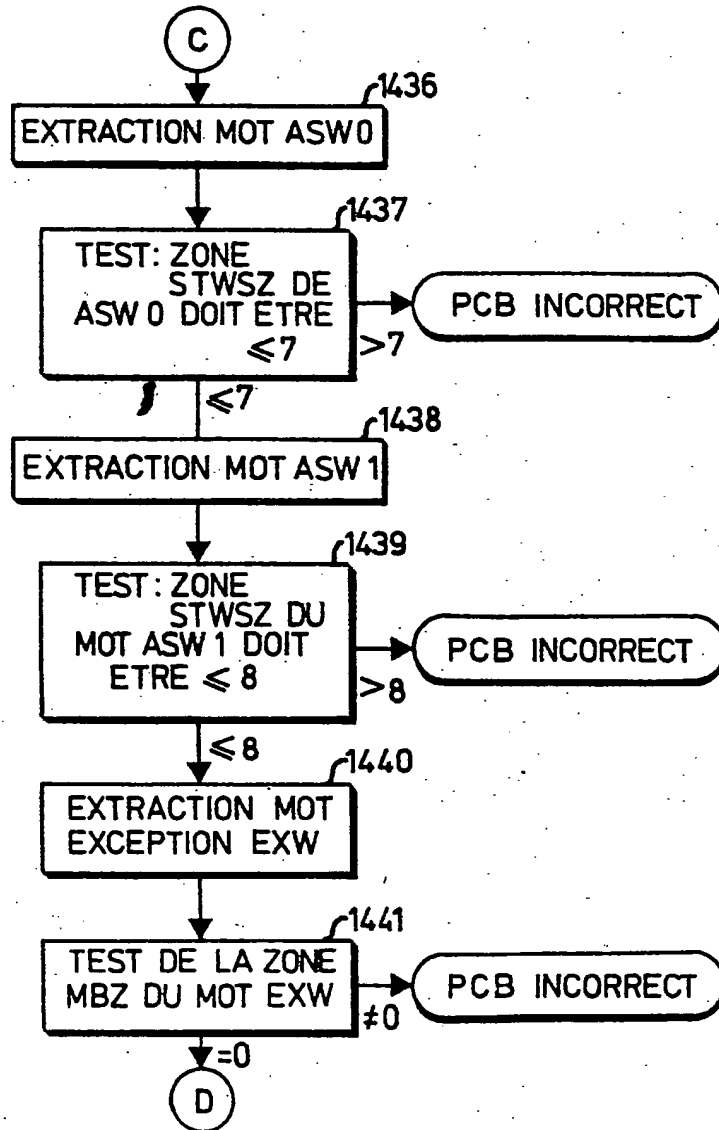


FIG. 14d

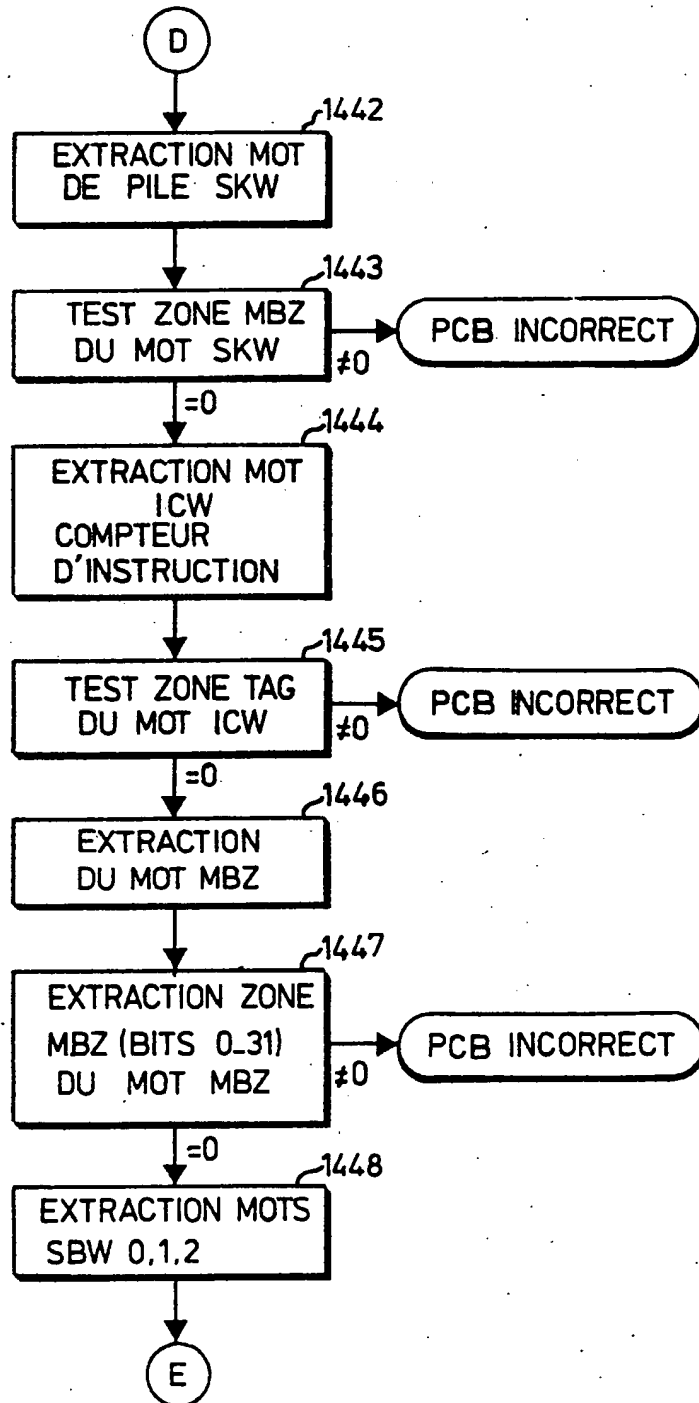


FIG. 14e

XXXII

2269150

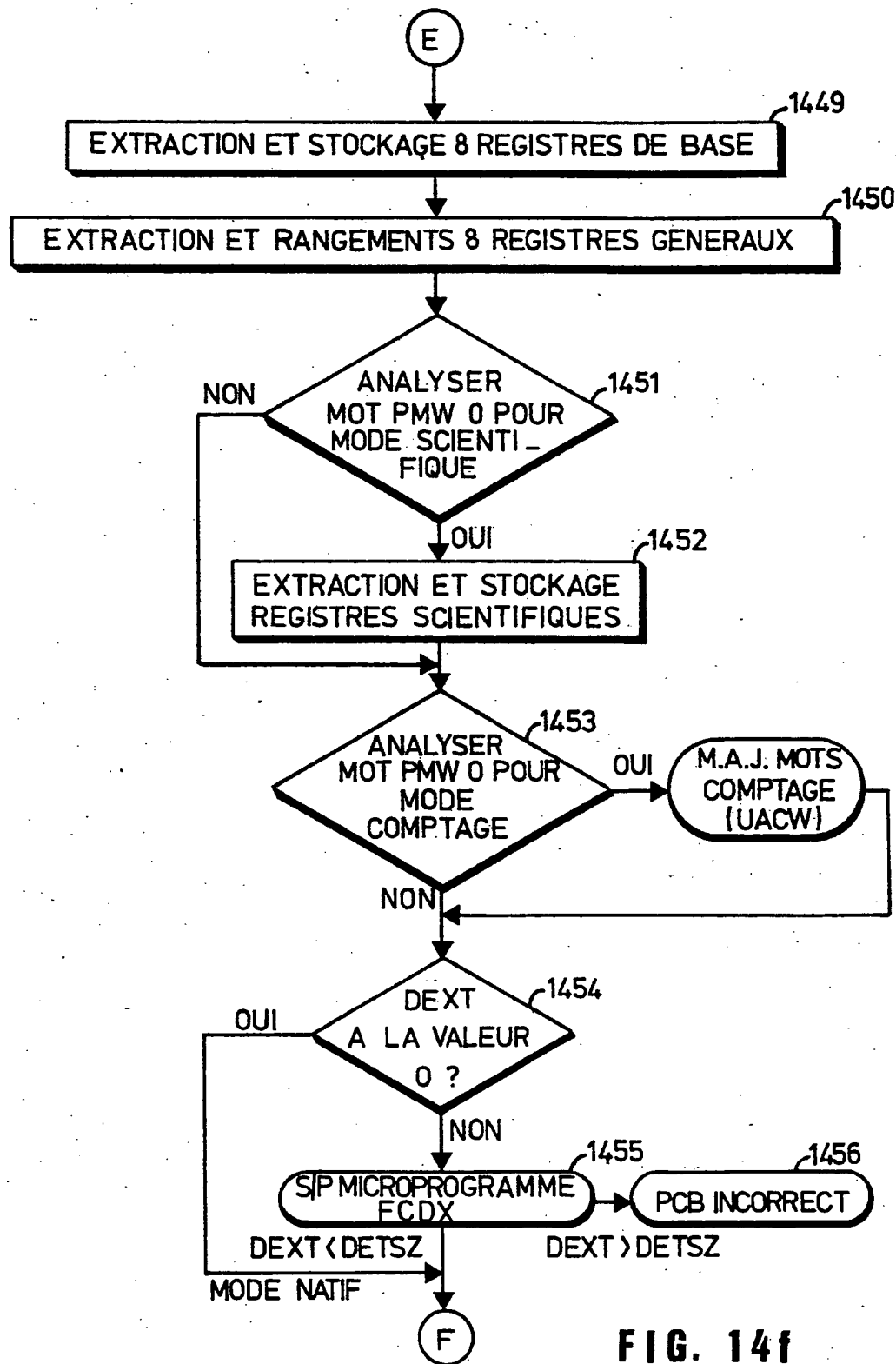


FIG. 14f

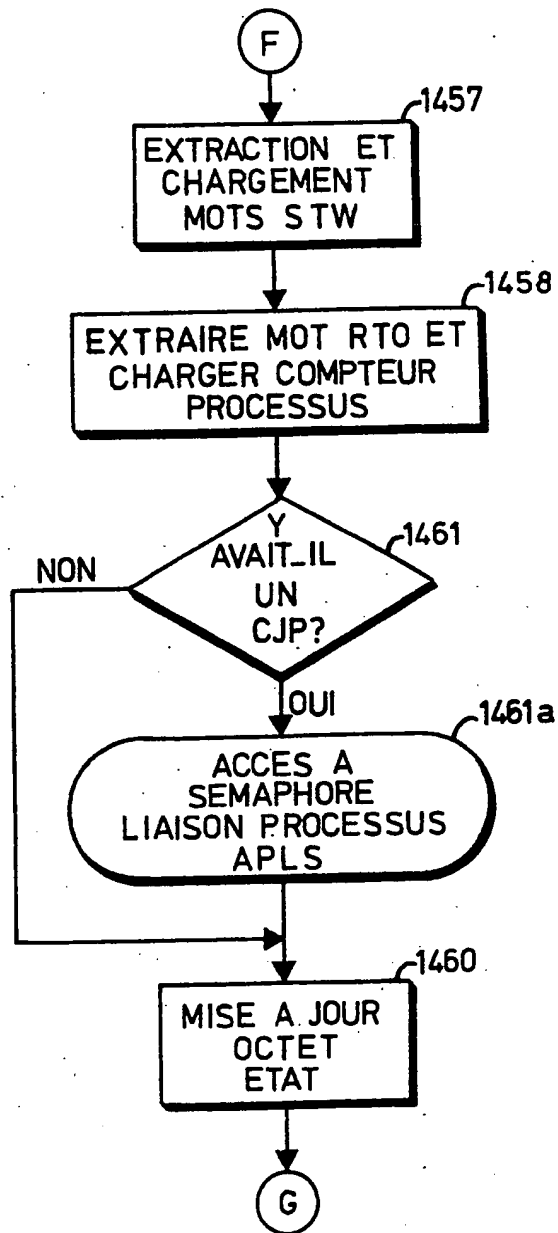


FIG. 14g

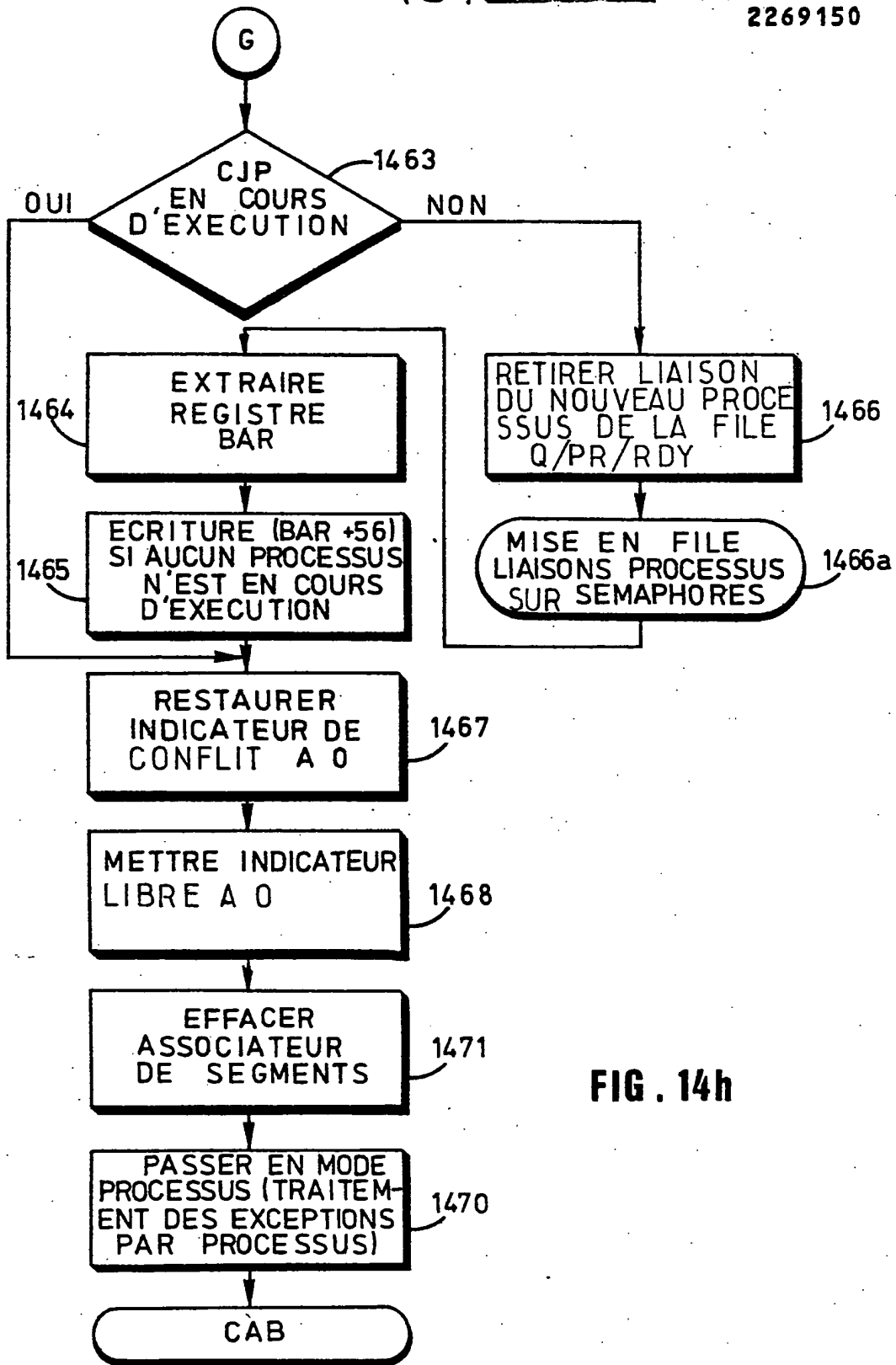


FIG. 14h

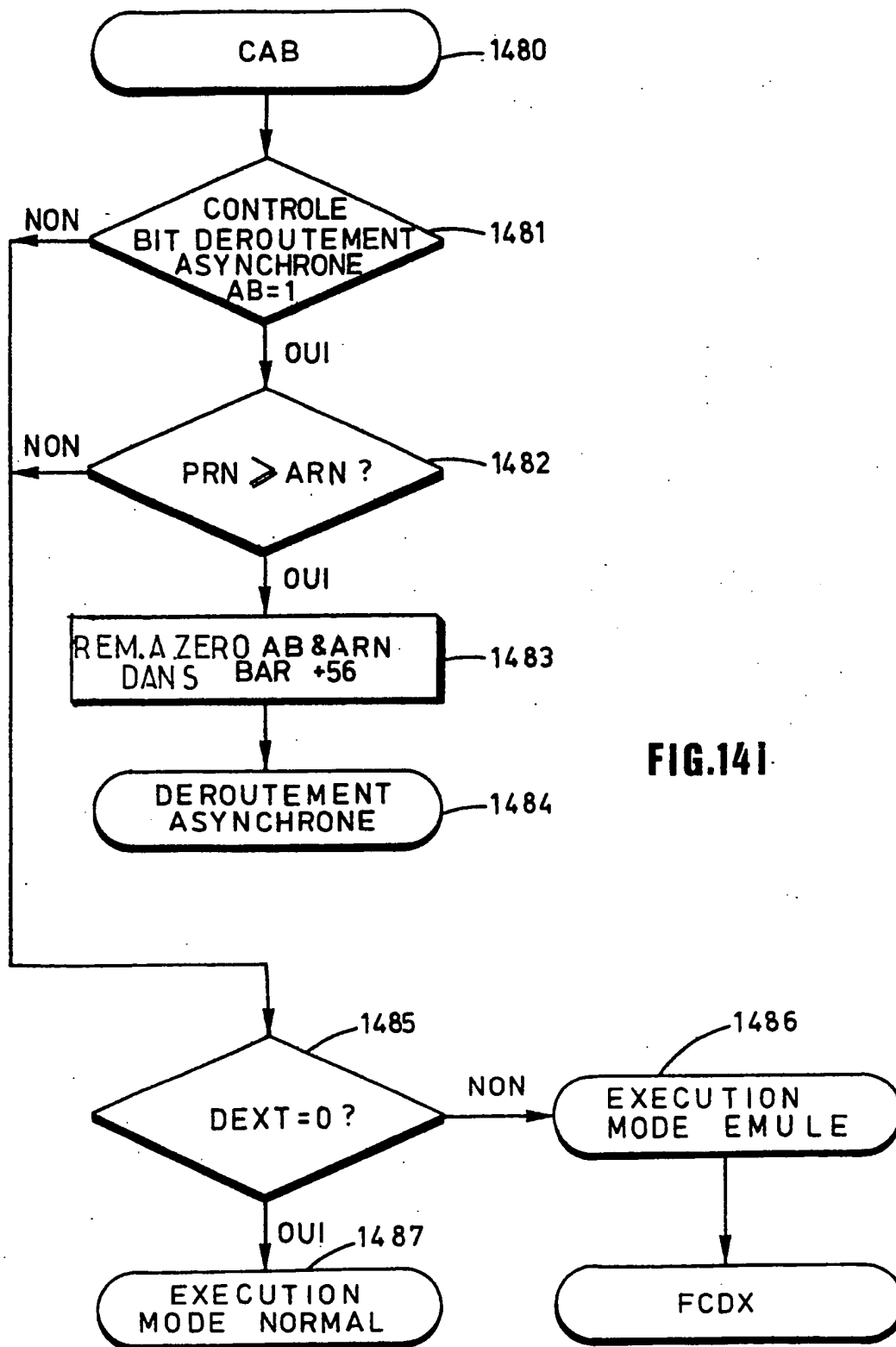


FIG.141